Indian Statistical Institute

Doctoral Dissertation

Statistical Monitoring of Image Data under Jump Regression Framework

A thesis submitted to the Indian Statistical Institute in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Statistics

Author: **Anik Roy**

Supervisor: **Dr. Partha Sarathi Mukherjee**

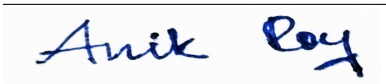
Interdisciplinary Statistical Research Unit (ISRU)

Statistical Sciences Division (StatSD)

Declaration of Authorship

I, Anik Roy, a Senior Research Fellow at Interdisciplinary Statistical Research Unit of Indian Statistical Institute, Kolkata, declare that this dissertation titled "Statistical Monitoring of Image Data under Jump Regression Framework" and the works presented in it are my own. I confirm the following statements.

- The work was done while in candidature for a research degree at Indian Statistical Institute, Kolkata.
- No part of this dissertation has been submitted for a degree or any other qualification at this institute or elsewhere.
- I have clearly cited all the previously published works that I have consulted for writing this dissertation.
- I have properly acknowledged all main sources of data used in this dissertation and the methods on which I have built my work.
- I have properly acknowledged all my co-authors who helped in developing different parts of this dissertation.



Anik Roy
Senior Research Fellow in Statistics
Interdisciplinary Statistical Research Unit
Indian Statistical Institute, Kolkata

Certificate from Supervisor

This is to certify that this doctoral dissertation entitled "Statistical Monitoring of Image Data under Jump Regression Framework," submitted by Anik Roy to Indian Statistical Institute is a record of bonafide research work under my guidance and supervision. The work contained in this dissertation is original and has not been submitted elsewhere for any degree.

Dr. Partha Sarathi Mukherjee
Associate Professor
Interdisciplinary Statistical Research Unit
Indian Statistical Institute, Kolkata

List of Publications

Published Research Articles

1. **Roy, A.** and Mukherjee, P.S. "Image Comparison Based On Local Pixel Clustering", *Technometrics*, 2024, 66(4), 495–506.
   DOI: 10.1080/00401706.2024.2322670. [Link]

2. Das, S., **Roy, A.** and Mukherjee, P.S. "Image registration for zooming: A statistically consistent local feature mapping approach", *Stat*, 2024, 13(1), e664. [Link]

3. **Roy, A.** and Mukherjee, P.S. "A control chart for monitoring images using jump location curves", *Quality Engineering*, 2024, 36(2), 439–452.
   DOI: 10.1080/08982112.2023.2232441. [Link]

4. Calderon-Garciduenas, L., Stommel, E.W., Torres-Jardon, R., Hernandez-Luna, J. Aiello-Mora, M., Gonzalez-Maciel, A., Reynoso, R., Perez-Guille, B., Silva-Pereyra, H., Tehuacanero-Cuapa, S., Rodriguez-Gomez, A., Lachmann, I., Galaz-Montoya, C., Doty, R.L., **Roy, A.**, and Mukherjee, P.S. "Alzheimer and Parkinson diseases, frontotemporal lobar degeneration and amyotrophic lateral sclerosis overlapping neuropathology start in the first two decades of life in pollution exposed urbanites and brain ultrafine particulate matter and industrial nanoparticles, including Fe, Ti, Al, V, Ni, Hg, Co, Cu, Zn, Ag, Pt, Ce, La, Pr and W are key players. Metropolitan Mexico City health crisis is in progress", *Frontiers in Human Neuroscience*, Section: Brain Health and Clinical Neuroscience, 2024, 17:1297467. [Link]

Submitted Research Articles

1. "Upper Quantile-based CUSUM-type Control Chart for Detecting Small Changes in Image Data" with Mukherjee, P.S.

2. "Rotation and Translation Invariant Monitoring of Shape and Size in Image Data" with Mukherjee, P.S.

3. "Adaptive Blind Image Deblurring" with Kang,Y. and Mukherjee, P.S.

Abstract

Image monitoring is an important research problem that has wide applications in various fields, including manufacturing industries, satellite imaging, medical diagnostics, and so forth. This problem, however, presents a challenging big data issue in the sense that, (i) it is characterized by high velocity and high volume of the data streams, (ii) observed image intensity functions are discontinuous in nature, have spatial structures, and it often contains noise, (iii) a typical grayscale image has a large number of pixels, implying high-dimensional nature of the data, (iv) in some applications, image surface often contains artifacts and insignificant anomalies (e.g., shadows, clouds, etc.), (v) sequence of images are often not geometrically aligned. In this dissertation, image monitoring schemes are developed on the basis of image intensity values, edges, and other complex features from the image surface. This dissertation aims to bridge the gap between the research fields of image processing and statistical process control and effectively address all the aforementioned issues. Our proposed methods in this dissertation make use of various state-of-the-art techniques from both research domains and help the research field of image monitoring stride forward. Numerical examples and statistical properties show that the proposed image comparison and monitoring methods in this dissertation perform well in various real-life scenarios. Furthermore, the novel methodological advancements proposed in this dissertation will be highly beneficial to the practitioners in various fields.

Acknowledgements

I am deeply grateful to the Indian Statistical Institute, Kolkata (ISI) for providing me with the research fellowship, an excellent academic environment, and the necessary infrastructure for my research. The exceptional computing facilities and extensive journal subscriptions offered by ISI were instrumental in enabling me to complete my research work within the stipulated time frame.

I would like to convey my heartfelt gratitude to my thesis advisor, Prof. Partha Sarathi Mukherjee. His boundless enthusiasm, active involvement, and insightful suggestions have profoundly influenced my dissertation and academic growth. This dissertation is the result of his invaluable guidance and our collaborative effort.

I would also like to thank Prof. Subir Kumar Bhandari, Prof. Sourabh Bhattacharya, and Prof. Kiranmoy Das for their insightful suggestions and unwavering motivation. Their guidance has been instrumental in fostering my passion for research. I would like to express my gratitude to Prof. Yicheng Kang, Mr. Sujay Das, Mr. Subhasish Basak, and Mr. Sagar Ghosh for their invaluable contributions to our collaborative research. I am immensely grateful to Prof. Tapas Samanta, Prof. Probal Chaudhuri, Prof. Arijit Chakrabarti, and Prof. Soumendu Sundar Mukherjee for their exemplary teaching during my PhD coursework. I extend my heartfelt appreciation to all the professors of the Statistical Sciences Division for their invaluable assistance and support in various matters that were crucial for the completion of this dissertation. Additionally, I would like to thank the staff of ISRU for their cooperation in providing a smooth work environment.

I dedicate this thesis to my parents. I would like to acknowledge them for their unwavering support, giving me the space and freedom to pursue this long journey of PhD program. Their sacrifice and understanding, which allow me to focus fully on my work, have been invaluable. I am also thankful to my elder brothers ('Dada' & 'Chor-Dada') for their constant motivations during my PhD days.

Now, I would like to thank my extended family. I have been encouraged, inspired, and tolerated by the incredible group of friends anyone ever had. It is my pleasure to thank Monitirtha, Meghna Di, Sayan, Sayantan, Biswadeep, Upama, Javed Da, Soutik,

Chirayata, Rahul Da, Sampurna, Annesha, Anirban, Mriganka, Amarnath Da, Arnab and Saptangshu for making my life far more joyful and relaxed. The unforgettable non-academic conversations over afternoon tea (at Lakkhan-da's place) and evening snacks (at Sanjeeb's stall) will always hold a special place in my heart. During my PhD journey, I had a nice stay with Rajib Da, Ayan, and Manabendra. Every morning used to start with tea prepared by Ayan, accompanied by lively discussion about the next mutton-treat, often initiated by the ever-enthusiastic Rajib Da. I also convey my wholehearted love to Jhelum, Priti and Suparna for the joyful hang out in the hostel. I must acknowledge Debraj and Debomitra for being beside me. Finally, the most heartfelt acknowledgment belongs to my nearest and dearest friend, Debika. Thank you for keeping me motivated with your generous emotional support.

I extend my gratitude to the chairs of our Research Fellow Advisory Committee (RFAC) during my PhD tenure, Prof. Bimal Roy and Prof. Ayanendranath Basu, as well as the conveners, Prof. Partha Sarathi Mukherjee and Prof. Abhik Ghosh, for their invaluable guidance and helpful suggestions. I thank the PhD-DSc committee, especially Prof. Rituparna Sen and Prof. Kiranmoy Das, for helping whenever needed. Additionally, I sincerely thank the Deans of Studies during my PhD tenure — Prof. Debasis Sengupta, Prof. Saurabh Ghosh (late), Prof. Amita Pal (Officiating Dean), Prof. Gopal Krishna Basak, and Prof. Biswabrata Pradhan (Present) — and the Director of the Indian Statistical Institute, Prof. Sanghamitra Bandyopadhyay, for their unwavering support in promoting research fellows. Their provision of necessary funds for attending various conferences worldwide has allowed us to gain exposure to cutting-edge research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Easy access to modern image acquisition techniques is making "images" a popular data format across various disciplines of science. Applications in manufacturing industries include stress and strain analysis of products, anomaly detection in the rolling process, steel and tile surface monitoring, and so forth. In medical science, various image modalities such as X-ray, CT-scan, MRI, and fMRI, are being widely used for medical diagnosis. In recent years, satellite images are increasingly being used in earth surface surveillance. It has become a basic tool for studying agriculture, forest science, ecology ecosystems, and many more. Note that in all these applications, sequential monitoring of images is important, making it a vibrant research area within the statistics community. For demonstration, Figure 1.1 shows three images of the Aral sea area in Central-Asia taken in 2000, 2010 and 2018, respectively. These three images clearly show the shrinkage of the lake over these 19 years of time. In all such applications, one important step is to *compare* images of the same scene captured at different times. There are certain challenges to

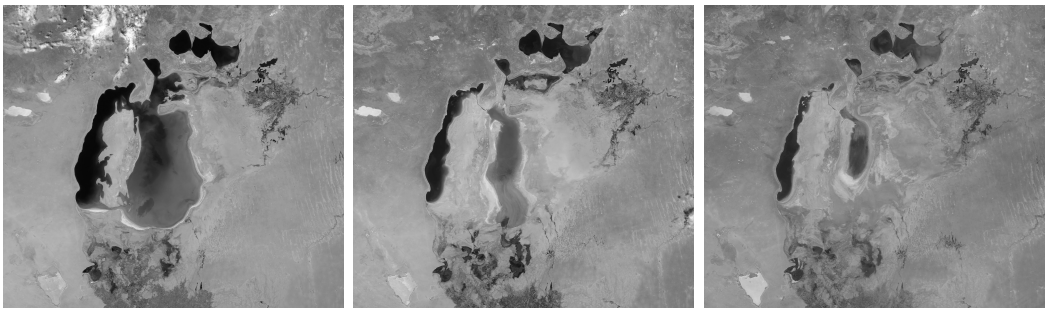


Figure 1.1: Three satellite images of Aral sea area taken on 25.08.2000 (left), 26.08.2010 (middle), and 21.08.2018 (right).

compare and monitor such images. For example, (i) the images may be not geometrically aligned perfectly, (ii) the differences between the first and the last image are very small in some regions, (iii) brightness and contrasts of various sub-parts of such images may be

different due to presence of clouds and many other reasons. This dissertation addresses many such issues while developing various image comparison and monitoring methods.

Next, we introduce the concepts of digital grayscale image, jump regression analysis, statistical process control and present a number of basic details relevant to this dissertation. Subsequently, we discuss the concept of sequential image monitoring, and various imaging issues related to this task, and present a brief literature review on image monitoring. Brief overview of this dissertation along with its contributions and novelty concludes Chapter 1.

1.1 Digital Image

A gray-scale image can be represented by a bivariate function $f(x, y)$, where function value $f(x, y)$ is associated to the brightness at the specific location (x, y) within the image. In literature, this function $f(x, y)$ is frequently referred to as the image intensity function. In order for a computer to process an image, it must be digitized in terms of both spatial location and brightness levels. In practical scenarios image acquisition devices, such as cameras and scanners, come with a digitizer that converts the captured images into digital formats. Throughout this dissertation, unless specified otherwise, all references to images are to gray-scale digital images.

A conventional 2-D digital gray-scale image can be represented by a matrix

$$\{f(i, j), i = 1, \ldots, n_1, j = 1, \ldots, n_2\}$$

where i and j are the indices of the rows and columns, respectively, as shown below:

$$\begin{pmatrix} f(1,1) & f(1,2) & \cdots & f(1,n_2) \\ f(2,1) & f(2,2) & \cdots & f(2,n_2) \\ \vdots & \vdots & \ddots & \vdots \\ f(n_1,1) & f(n_1,2) & \cdots & f(n_1,n_2) \end{pmatrix}$$

Each element of the matrix is referred to as a *pixel*. The image *resolution* is determined by n_1 and n_2: a higher resolution corresponds to larger values of n_1 and n_2, while a lower resolution corresponds to smaller values. For the pixel at position (i, j), the digitized value of $f(i, j)$ is typically an integer within the range $[0, L-1]$, where 0 represents black and $L-1$ represents white. The value of $f(i, j)$ indicates the shade of the image at that position and is often referred to as the gray level of the pixel. For more detailed information, refer to Gonzalez and Woods (2019); Qiu (2005).

1.2 Jump Regression Analysis

Conventional nonparametric regression methods are appropriate under the assumption that the regression functions are continuous in nature. In the presence of jumps in the regression function, estimates produced by conventional methods are not statistically consistent at the jump positions [Qiu (2007)]. The intensity function of an image may have discontinuities at object boundaries. Given that an image can be seen as a jump surface of the image intensity function, edge detection and image restoration problems in image processing are closely related to jump regression problems in statistics.

In literature, when the true regression function is assumed to have jumps in the design space, the corresponding regression analysis is commonly known as *jump regression analysis* (JRA). One significant application of JRA model includes the modeling of gray-scale image surfaces. More specifically, a gray-scale image can be expressed by a two-dimensional regression model [Qiu (2005)] as follows:

$$w_{ij} = f(x_i, y_j) + \varepsilon_{ij}, \quad i = 1, 2, \ldots, n_1; \quad j = 1, 2, \ldots, n_2, \tag{1.2.1}$$

where in the context of digital images, x_i denotes the i-th row of the image, y_j denotes its j-th column, f is the image intensity function, $f(x_i, y_j)$ is the true image gray level at the (i, j)-th pixel, ε_{ij} denotes the noise at the (i, j)-th pixel, and w_{ij} is the observed image gray level at the (i, j)-th pixel. Note that, the image intensity function f has jumps at the outlines of the image objects which is popularly known as Jump Location Curves (JLCs). In literature, positions at which f have jumps are called *step edges*, and positions at which the first-order derivatives of f have jumps are called *roof edges* [Joo and Qiu (2009)].

1.3 Statistical Process Control

In today's computer age, the statistical monitoring of sequential observations has become a significant research area. When it comes to sequentially monitoring a longitudinal process, a major statistical tool is *Statistical Process Control* (SPC). For a systematic description of the basics of SPC, see books Hawkins and Olwell (2012); Montgomery (2007); Qiu (2013). Typically, control charts in SPC are extensively utilized to monitor the stability of sequential processes across various fields, including manufacturing industries, ecological studies, and healthcare systems [Qiu (2024, 2019)]. The SPC charts generally assume that there are two sources of variability in the process measurements: a "common cause" that stems from inevitable randomness, and a "special cause" that arises when an undesirable

variability, such as mechanical defects, improper machine handling, human error, the onset of specific medical conditions, etc., interferes. SPC of a process is generally divided into four phases: Phase 0, Phase I, Phase II and Phase III [Megahed et al. (2010); Vining (2009)]. However, majority of the literature from SPC focus on Phase I and Phase II only. Phase 0 primarily involves data gathering and determines the key characteristics of the process control. The primary objective of Phase I is to adjust the process so that it can run in a stable manner [Chakraborti et al. (2008); Mukherjee (2023)]. This control-and-adjustment process typically needs to be repeated multiple times until it is confirmed that the process is in-control (IC). Once this is achieved, for the online monitoring, a Phase II control chart can be properly designed using an IC dataset collected after Phase I SPC. The major goal of Phase II monitoring is to identify any significant changes in the process distribution occurring after an unknown time point. Additionally, Phase III focuses on ongoing model maintenance, such as updating the model and control limits over time [Zwetsloot et al. (2024)]. In certain SPC applications (e.g., satellite image monitoring, disease risk monitoring, etc..) it becomes difficult to adjust the process in Phase I. In those applications, the Phase I step is not relevant. In this dissertation, we mainly consider Phase II monitoring schemes.

1.3.1 Traditional SPC charts

Shewhart control chart: Assume X is a single quality variable in a specific process monitoring scenario; it is a continuous numerical variable with an in-control (IC) distribution of $N(\mu_0, \sigma^2)$. The observations obtained at the nth time point are denoted as:

$$X_{n1}, X_{n2}, \ldots, X_{nm},$$

where $m \geq 2$ represents the batch size. To determine whether the process is IC at the n-th time point it is natural to use z-test. The process is considered as out-of-control (OC) if

$$\overline{X}_n > \mu_0 + Z_{1-\alpha/2}\frac{\sigma}{\sqrt{m}} \quad \text{or} \quad \overline{X}_n < \mu_0 - Z_{1-\alpha/2}\frac{\sigma}{\sqrt{m}},$$

where X_n is the sample mean of $\{X_{n1}, X_{n2}, \ldots, X_{nm}\}$, and $Z_{1-\alpha/2}$ is the $(1-\alpha/2)$-th quantile of the $N(0,1)$ distribution. In practice, μ_0 and σ could be unknown and need to be estimated from an IC dataset $\{(X^*_{i1}, X^*_{i2}, X^*_{i3}, \ldots X^*_{im}) : i = 1, 2, \ldots, M, \}$. Let $\overline{X}^*_i$ and R^*_i be be the sample mean and sample range of the i-th batch of IC dataset and $\overline{\overline{X}}^*$ and $\overline{R}^*$ be the sample average of $\{\overline{X}^*_i := 1, 2, \ldots, M\}$ and $\{R^*_i := 1, 2, \ldots, M\}$, respectively. It is easy to verify that $\overline{\overline{X}}^*$ and $\overline{R}^*/d_1(m)$ are the unbiased estimator for μ_0 and σ, respectively. Note that $d_1(m)$ is a constant that depends on the value of m. Then Shwewhart control

chart [Shewhart (1931)] raise a signal at the n-th time point if,

$$\overline{X}_n > \overline{\overline{X}}^* + Z_{1-\alpha/2}\frac{\overline{R}^*}{d_1(m)\sqrt{m}} \quad \text{or} \quad \overline{X}_n < \overline{\overline{X}}^* - Z_{1-\alpha/2}\frac{\overline{R}^*}{d_1(m)\sqrt{m}}.$$

CUSUM control chart: At a given time time point, the Shewhart chart determines whether a process is in-control solely based on the observed data at that specific time point. This approach is often inadequate for Phase II process monitoring because past observed data can provide valuable insights about the current process performance. To address this limitation, Page (1954) introduced the CUSUM chart, and many different CUSUM charts have been suggested in the literature for various purposes [Hawkins (1987); Hawkins and Olwell (2012)]. Here, we briefly describe the fundamental CUSUM chart for detecting a mean shift of a normal-distributed process. Let us assume that the in-control process distribution is $N(\mu_0, \sigma^2)$, and the process observations for online monitoring are $\{X_n, n = 1, 2, \ldots\}$. Then, the CUSUM chart statistics for detecting a mean shift are defined by

$$C_n^+ = \max(0, C_{n-1}^+ + \frac{X_n - \mu_0}{\sigma} - k), \tag{1.3.1}$$

$$C_n^+ = \max(0, C_{n-1}^+ + \frac{X_n - \mu_0}{\sigma} - k), \quad \text{for } n \geq 1, \tag{1.3.2}$$

where $C_0^+ = C_0^- = 0$, and $k > 0$ is an allowance constant. The chart raises a signal when

$$C_n^+ > c \text{ or } C_n^- < -c, \tag{1.3.3}$$

where $c > 0$ is a control limit. In the above CUSUM chart (1.3.1)-(1.3.2), the allowance constant k is typically pre-specified. The control limit c is then chosen to ensure that the in-control *average run length* (ARL), denoted as ARL_0, meets a specified value. The ARL_0 is represented as the average number of observations from the beginning of process monitoring to a signal when the process is in-control. From (1.3.1)-(1.3.2), it is evident that the charting statistics C_n^+ and C_n^- make use of the all available data before n-th time point, and they re-start from 0 again when the cumulative information suggests no significant evidence of a mean shift in the sense that:

$$C_{n-1}^+ + \frac{X_n - \mu_0}{\sigma} < k \text{ and } C_{n-1}^- + \frac{X_n - \mu_0}{\sigma} > -k$$

It is important to note that the CUSUM chart's restarting mechanism offers a significant theoretical advantage in terms of minimizing detection delay. According to Moustakides (1986), the CUSUM chart with an allowance constant k has the shortest out-of-control

ARL (ARL_1) value among all charts with a fixed ARL_0 value for detecting a persistent shift of size $\delta = 2k$.

EWMA control chart: Despite their strong theoretical properties, CUSUM charts were challenging to use in the 1950s due to the lack of computers. As an alternative, a simpler chart called the exponentially weighted moving average (EWMA) chart was proposed by Roberts (1959). Under the similar assumptions and notations of CUSUM chart in (1.3.1)-(1.3.2), the EWMA charting statistic is defined as

$$E_n = \lambda X_n + (1-\lambda)E_{n-1}, \tag{1.3.4}$$

where $E_0 = \mu_0$, and $\lambda \in (0, 1]$ is a weighting parameter. To effectively use the EWMA chart in practice, it is essential to select appropriate λ values. Generally, small λ values are suitable for detecting relatively small mean shifts, while large λ values are better for detecting relatively large mean shifts. From (1.3.4), it can be shown easily that,

$$E_n = \lambda \sum_{i=1}^{n} (1-\lambda)^{n-i} X_i + (1-\lambda)^n \mu_0, \tag{1.3.5}$$

and when the process is in-control up to the current time point n, we have

$$E_n \sim N\left(\mu_0, \frac{\lambda}{2-\lambda}\left[1-(1-\lambda)^{2n}\right]\sigma^2\right). \tag{1.3.6}$$

From (1.3.5) it is clear that E_n is a weighted average of μ_0 and all available observations up to n, and the weight received by X_i decays exponentially fast when i moves away from n. Then, based on the expression in (1.3.6), the EWMA chart gives a signal of process mean shift when

$$|E_n - \mu_0| > c\sigma\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2n}]}, \tag{1.3.7}$$

where $c > 0$ is a control limit. In the EWMA chart (1.3.7), with a pre-specified value of the the weighting parameter λ, the control limit c is chosen such that a given ARL_0 value is achieved.

Change point based control chart: One of the major disadvantage to use the Shewhart, CUSUM, and EWMA charts described earlier, is that the in-control parameters μ_0 and σ need to estimated in advance, which can be impractical for some applications. To address this issue, Hawkins et al. (2003) propose a *change point detection* (CPD) based control chart, outlined below. For process observations $X_1, X_2, \ldots, X_n$, it is assumed that

they follow the following change-point model:

$$X_i = \begin{cases} \mu_0 + \varepsilon_i, & \text{if } i = 1, 2, \ldots, \tau, \\ \mu_1 + \varepsilon_i, & \text{if } i = \tau + 1, \tau + 2, \ldots, n, \end{cases} \tag{1.3.8}$$

where τ is a change-point, and $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ is a sequence of i.i.d. random variables with the common distribution $N(0, \sigma^2)$. Then, the likelihood ratio test statistic for testing the existence of a change-point is defined as

$$T_{\max,n} = \max_{1 \le j \le n-1} \sqrt{\frac{j(n-j)}{n}} \left| \frac{X_j - X'_j}{\widetilde{S}} \right|, \tag{1.3.9}$$

where X_j and X'_j are sample means of the first j and the remaining $(n-j)$ observations in $\{X_1, X_2, \ldots, X_n\}$, respectively, and

$$\widetilde{S}_j^2 = \sum_{i=1}^{j} (X_i - X_j)^2 + \sum_{i=j+1}^{n} (X_i - X'_j)^2.$$

The CPD chart gives a signal of mean shift when

$$T_{\max,n} > c_n \tag{1.3.10}$$

where $c_n > 0$ is a control limit that may depend on n. After a signal is given, an estimate of the change-point τ is given by the maximizer found in (1.3.9).

Here, we outline four fundamental control charts for monitoring process mean shifts when the process observations are univariate. In literature, there are numerous variations of each type aimed at detecting changes in process mean, variance, and other aspects of the process distribution. For more information, see references such as Holland and Hawkins (2014), Zou and Tsung (2011), Zou and Tsung (2010), Hawkins and Deng (2010), Zamba and Hawkins (2006), Woodall and Ncube (1985), and many more.

1.4 Sequential Monitoring of Images

In certain applications, image data are collected as data streams, in the sense that new images are continuously acquired from a longitudinal process over time (e.g., fMRI images or images from a rolling process). In such applications, a fundamental task is to monitor the sequence of images to detect any significant changes in the underlying longitudinal process. This is known as the image monitoring problem. Thus, both image comparison and image monitoring are critically important in many applications. In manufacturing

industries, applications include stress and strain analysis of products, anomaly detection of rolling processes, inspection of composite material fabrication, quality control in liquid crystal display manufacturing, structural health monitoring, and so forth. In the field of medical imaging, X-ray, CT-scan, MRI, and fMRI have been widely used for medical diagnosis. Nowadays, satellite images have become a basic tool in surveillance of the earth's surface. It has been widely used in the research of agriculture, forest science, ecology and ecosystem, coastal resources, environment, etc. A security related example involves airspace image monitoring performed by the military to detect foreign air crafts. Since 1972, USGS in collaboration with NASA launched 9 satellites to get images of earth's surface which is very well known as Landsat Project. Data from this project are easily available for researchers all around the world. Monitoring these sequence of images have received attention in different scientific disciplines. However, monitoring image data presents a challenging big data issue in the sense that,

1. It characterized by high velocity and high volume of the data streams
2. Image surface contains discontinuities and other singularities, therefore conventional smoothing techniques are ineffective
3. Image data is high-dimensional: a typical grayscale image has a huge number of pixels . In practice, changes in the image involve a relatively small number of pixel coordinates on the image surface. In the literature, this phenomenon is known as sparsity.
4. In satellite image monitoring, image surface often contains artifacts and insignificant anomalies (e.g., shadows,clouds, etc.). It is necessary to disregard them during the process of monitoring.
5. Often in image monitoring applications, sequence of images are not geometrically aligned. For reliable monitoring, it is necessary to aligned them first. However, it poses a significant challenges due to its computational demand at each subsequent stage of image moinitoring.

1.5 Imaging Issues Related to Monitoring

Images obtained from a process often contain various forms of contamination, including geometric misalignment, pointwise noise, and spatial blur. To make subsequent image monitoring more reliable, proper processing of the observed image surface is critically important. This section focuses on some fundamental image processing issues related to image process monitoring.

Image Denoising: Image denoising is an important task in image analysis. It is often required to pre-process images so that subsequent image monitoring becomes more reliable. Besides noise removal, it is essential for an image denoising procedure to preserve the complex structure of the image surface, such as edges. In JRA framework, image denoising can be achieved by estimating a discontinuous surface from noisy data, because a monochrome image can be regarded as a surface of the image intensity function and such a surface has jump discontinuities at the boundary of image objects [Qiu (2009); Qiu and Mukherjee (2010, 2011); Mukherjee and Qiu (2013)]. In the literature, there are several existing procedures for image denoising and restoration. One popular approach is based on Bayesian estimation, using Markov random field (MRF) modeling and maximum a posteriori (MAP) algorithm [Besag (1986); Geman and Geman (1984); Fessler et al. (2000); Godtliebsen and Sebastiani (1994)]. Closely related methods employ a regularization approach, by minimizing a certain objective function that enforces a roughness penalty in addition to a term measuring the fidelity of an estimator to the data [Rivera and Marroquin (2002); Marroquin et al. (2001)]. Another alternative denoising method is based on adaptive weights smoothing, as detailed in a series of papers by Joerg Polzehl and his co-authors [Polzehl and Spokoiny (2000)].

Image Registration: In applications of image comparison and monitoring, acquired images from the process are often geometrically mismatched, because the relative positions between the camera and image objects cannot be exactly the same at different times in practice. Therefore, for a reliable and meaningful image comparison the images should be geometrically matched up first [Feng and Qiu (2018); Qiu (2018)]. In image processing literature, this is popularly known as *image registration* [Avants et al. (2008)].

$$\begin{aligned} w_R(x_i, y_j) &= f_R(x_i, y_j) + \varepsilon_R(x_i, y_j), \\ w_M(x_i, y_j) &= f_M(x_i, y_j) + \varepsilon_M(x_i, y_j), \quad i, j = 1, \cdots, n, \end{aligned} \tag{1.5.1}$$

where $\{(x_i, y_j)\}$ are equally-spaced pixels, $f_M(x_i, y_j)$ and $f_R(x_i, y_j)$ are the true moved and reference image intensity functions, $w_R(x_i, y_j)$ and $w_M(x_i, y_j)$ are their observed versions, and $\{\varepsilon_M(x_i, y_j)\}$ and $\{\varepsilon_R(x_i, y_j)\}$ are i.i.d. random errors with mean 0 and unknown variances. Additionally, for a transformation $\mathbf{T}(x, y) = (T_1(x, y), T_2(x, y))$, it is assumed that reference image and moved image has the following relationship:

$$f_M(T_1(x, y), T_2(x, y)) = f_R(x, y), \quad (x, y) \in \Omega = [0, 1] \times [0, 1], \tag{1.5.2}$$

The main objective of image registration is to estimate the transformation $\mathbf{T}(x, y)$. Based on the nature of transformation, the problem of image registration can be classified in two parts viz., rigid-body image registration and non-rigid-body image registration. Rigid-body transformation include rotation [Reddy and Chatterji (1996)] and translation

[Wolberg and Zokai (2000)], whereas, non-rigid-body transformation includes zooming, shearing, elastic transformation, and so forth [Das et al. (2024); Bay et al. (2006); Lowe (2004); Davatzikos et al. (1996)]. In most of the image surveillance applications where the geometric difference between $f_M(x, y)$ and $f_R(x, y)$ is mainly due to the position move between the products and the camera, it is reasonable to assume that $\mathbf{T}(x, y)$ is a rigid-body transformation defined as follows:

$$\begin{aligned} T_1(x, y) &= x \cos\theta + y \sin\theta + h \\ T_2(x, y) &= -x \sin\theta + y \cos\theta + k, \end{aligned} \tag{1.5.3}$$

Where θ is the rotation parameter and (h, k) are the translation parameters corresponding to the x-axis and y-axis, respectively. In the rigid-body image registration problem, our aim is to estimate the parameters (θ, h, k) from the observed images. See Qiu and Xing (2013); Xing and Qiu (2011) for more discussion regarding image registration.

Image Deblurring: The challenge of image denoising mainly deals with pointwise contamination on the image surface. However, in real-world scenarios, spatial blur is another form of image contamination that can occur due to various factors, such as the relative motion between the camera and the scene being captured. The primary goal of *image deblurring* is to retrieve the true signal from the blurred surface. In the literature, the blurring mechanism is described as follows:

$$w(x, y) = H\{f\}(x, y) + \varepsilon(x, y), \tag{1.5.4}$$

where $H\{f\}(x, y) = \int_{\mathbb{R}^2} h(u, v) f(x - u, y - v)\, dudv$ denotes the convolution of the *point spread function* (psf) h and the true image intensity function f, and $\varepsilon(x, y)$ represents the pointwise noise. The psf h describes the spatial contamination(i.e., blurring) of f in the imaging process. The problem of *image deblurring* seeks to estimate $f(x, y)$ from the observed blurred image surface $w(x, y)$. It is important to note that image deblurring is considered an "ill-posed" problem because multiple sets of h and f can correspond to the same w, even when there is no pointwise noise in w. So, in the early literature, h is often assumed to be known [Figueiredo and Nowak (2003)]. However, in practical scenarios, this assumption may not be realistic. For instance, satellite images are frequently blurred due to factors such as wind, atmospheric turbulence, optical system aberrations, and others, making it challenging to define h precisely. More recent research in this field aims to estimate $f(x, y)$ when h is not completely known, a problem referred to as *blind image deblurring* [Joshi and Chaudhuri (2005); Hall and Qiu (2007a,b)]. See [Kang (2020); Kang et al. (2018); Qiu and Kang (2015)] for more recent development on blind image deblurring.

1.6 Literature Review on Image Monitoring

In nonparametric regression literature, preliminary studies on this problem have started using regression curve comparisons [Hall and Hart (1990); King et al. (1991); Kulasekera (1995); Dette and Munk (1998a,b); Wang and Ye (2010)]. Readers are referred to Dette and Neumeyer (2001) for a good review on comparing regression curves. The central ideas of nonparametric comparison of image surfaces are based on nonparametric ANCOVA with multiple covariates. Bowman (2006) suggests a generalization of ANOVA-type test for comparing regression surfaces. A more flexible test based on $\mathcal{L}^2-$ distance between regression surface estimate is proposed by Wang and Ye (2010). These existing methods are the generalizations of the method proposed by Dette and Neumeyer (2001). Recently, Zhao et al. (2020) suggest a testing procedure for comparing nonlinear curves and surfaces in the usual context of semiparametric regression. Since images have been traditionally used as a form of data in the chemical and manufacturing processes, there is an extensive discussion in the statistical process control (SPC) literature [Megahed et al. (2011)]. Most of the existing methods consist of the following two steps. Firstly, extract important image features from the observed images and use a conventional univariate "or" multivariate control chart to get signals from process shifts. However, a number of existing methods select a set of pre-specified regions known as regions of interest (ROIs) for individual images and construct control charts based on an appropriate summary statistic (e.g: average intensity) for ROIs over time using conventional generalized likelihood ratio (GLR) control chart by Megahed et al. (2012) or multivariate generalized likelihood ratio (MGLR) chart by He et al. (2016). The GLR control chart by Megahed et al. (2012) is designed primarily for monitoring specific patterns in images but struggles to detect multiple faults on image surfaces. The MGLR chart is more suitable for identifying multiple faults but the control chart statistic contains the variance-covariance matrix which is not invertible in high dimensional applications. Okhrin et al. (2020) address this issue and describe a more sophisticated version of the GLR method to get rid of this problem. Amirkhani and Amiri (2020) partition the whole image into a set of specific regions and for comparing with the nominal image, they use a test based on one-way ANOVA ,and then construct a p-value based control chart to detect the out-of-control signal in an image sequence as soon as possible. Koosha et al. (2017) propose profile monitoring approach for monitoring. They have taken the wavelets coefficients as a frequency domain feature and monitored the change in the coefficients to keep track of images. See Okhrin et al. (2019) for more discussion about different methods based on frequency domain features. There is also a substantial number of research work focusing on anomaly detection while monitoring images for quality inspection [e.g., Yan et al. (2017)]. Bui and Apley (2018a, 2021) propose several approaches for monitoring images of textured surfaces. In the field of

additive manufacturing, image surveillance can help identify geometric nonconformities. For a detailed discussion, see Grasso and Colosimo (2017).

1.7 Brief Overview of The Dissertation

Inspired by the wide range of applications, my dissertation primarily concentrated around the continuous surveillance of image data from various disciplines. The monitoring schemes are developed on the basis of image intensity values, edges, and other complex features from the image surface. The dissertation is broadly divided into two main areas: (I) intensity-based image monitoring, and (II) feature-based image monitoring.

Monitoring image intensity poses a substantial challenge due to numerous factors: (i) the presence of discontinuities and singularities on the image surface renders conventional smoothing techniques somewhat useless; (ii) the high dimensionality of image data, given the considerable number of pixels in a typical grayscale image. Notably, changes in the image typically involve only a small subset of pixel coordinates, a feature called as sparsity in the literature. Additionally, within the context of statistical process control (SPC), it has been mounted that conventional multivariate control charts exhibit confined efficacy in signal detection when faced with high data volumes. In Chapter 2, we introduce a Cumulative Sum (CUSUM) control chart designed for the online monitoring of grayscale images while preserving the JLCs on the image surface. To cope with the sparsity assumption, the central idea involves constructing a global charting statistic based on the upper-q-quantile of local CUSUM statistics. The proposed algorithm is theoretically justified in terms of asymptotic optimality, and numerical comparisons with existing methods demonstrate its efficacy across various scenarios.

Within the existing literature, a predominant majority of methods rely on changes in image intensity to identify out-of-control images. However, intensity based monitoring often impractical in numerous real-life applications where alterations in contrast between the background and foreground should not flag an image as out-of-control, as long as the boundaries of image objects remain consistent. The rest of the dissertation delve into the surveillance of image sequence while ignoring insignificant alternation of image intensities.

In Chapter 3, we present a Shewhart-type control chart designed for the monitoring of grayscale images, utilizing detected edges. The fundamental concept involves monitoring the point set of detected edge pixels of the image sequence. This approach is particularly relevant in scenarios where smooth changes in background of the image object should not trigger false alarms, provided that the boundaries of image objects remain stable. The proposed monitoring procedure is designed to be easily implementable across a spectrum

of real-life applications. Numerical studies indicate its robust performance across various situations when compared with several competing methods.

One significant drawback of the proposed algorithm in Chapter 3 is that it only ignores smooth insignificant change. A more flexible algorithm is introduced in Chapter 4 that effectively disregards any type of insignificant anomalies while monitoring. The central idea is to compare two local neighborhood corresponding to the every pixel coordinate of the two surfaces. Since our goal is to compare significant JLCs, we employ pixel clustering in the respective neighborhoods and compare them using the *Variation of Information* metric. A major advantage of this method is that it does not require explicit edge detection. However, edge information is preserved by the local clustering algorithm, enabling it to effectively ignore insignificant JLCs during surveillance.

For reliable image comparison, *image registration* is an essential step. In the literature on image comparison and monitoring, it is typically assumed that images are properly aligned. However, misalignment is quite common, particularly in satellite imaging. Existing literature often utilizes image registration algorithms during the pre-processing stage, which can be computationally expensive and not always accurate. Therefore, an image monitoring scheme that inherently handles image registration is highly desirable. In Chapter 5, we introduce a shape and size monitoring algorithm that is invariant under rotation and translation and efficient in detecting changes in the shape and size of the image object. The main motivation behind the proposed method is the absence of an algorithm in the literature that handles image registration issue by its construction. Our approach involves constructing a test statistic, based on the distribution of distances from the centroid of the image object to its boundary, for comparing two shapes or images. To facilitate online monitoring, we propose a nonparametric CUSUM control chart based on the aforementioned statistics.

A few remarks and a number of future research directions presented in Chapter 6 conclude this dissertation.

1.8 Contributions and Novelty of This Dissertation

- This dissertation makes an effort to bridge the gap between research fields of image processing and statistical process control. Image monitoring research traditionally performed by the statisticians use various process control charts without using advanced image processing tools. Likewise, computer scientists use advanced machine learning techniques without using any advanced knowledge of statistics or process control charts. Our proposed methods in this dissertation make use of various

state-of-the-art techniques from both research domains, and help the research field of image monitoring stride forward.

- Chapter 2 makes use of an advanced image denoising technique and combine it with a version of CUSUM control chart for efficiently detecting a small change in an image. The proposed method shows a much improved performance compared to some its competitors.

- Chapter 3 proposes to monitor the *jump location curves* or the edges of he image objects rather than monitoring pixelwise intensity values. Major advantage of this approach is that it can ignore minor anomalies in the images that can occur due to various reasons. For example, while capturing a picture, some parts of an image can be overexposed or underexposed to light. Change of intensity values merely due to such reasons should not indicate a meaningful change in the image objects. The proposed method can handle such issues very well.

- Chapter 4 develops a method of image comparisons based on local pixel clustering, without explicitly detecting the edges. Based on a tuning parameter, the procedure can ignore parts of *jump location curves* where the jump sizes are small. Thus, it can ignore unimportant edges while performing image comparisons. The practitioner can choose the value of the tuning parameter based on the nature of the images and the purpose of the study. This image comparison method can be used for image monitoring in various settings.

- Chapter 5 aims to solve a long-standing problem of incorporating rigid-body transformations of the image objects while comparing or monitoring images. Application of the proposed method does not require the pre-processing step of *image registration*. Simplicity and effectiveness of the technique should attract attention from both the practitioners and researchers.

- Not only our proposed methods of image comparisons and image monitoring are useful in many practical applications, the adapted versions of the central ideas of our solutions can be used to develop many other techniques in applied statistics and machine learning.

Chapter 2

Intensity-based Image Monitoring: Upper-q-Quantile CUSUM Control Chart

2.1 Introduction

This chapter is mainly dedicated to intensity-based image monitoring. Here, we present an image monitoring control chart to detect subtle changes in the image intensity function in the presence of noise while accommodating the edges and other complicated structures within the images. Note that, it is indeed a challenging task because of the following

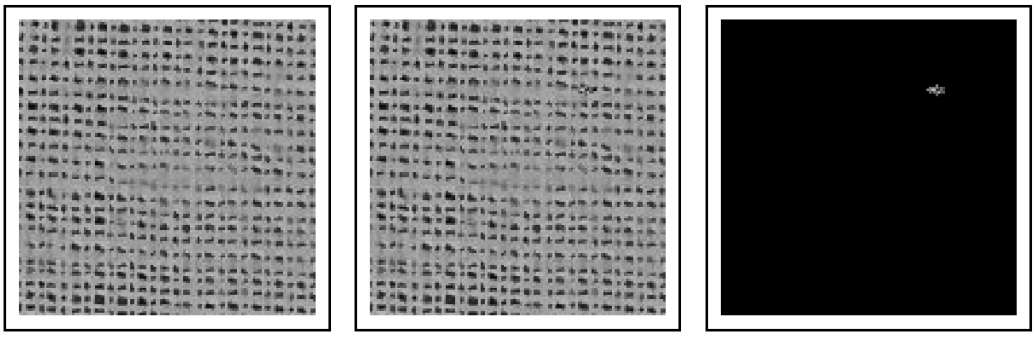


Figure 2.1: Example of a small anomaly at the top right corner of a textile image. (a) Image without anomaly. (b) Image with anomaly. (c) Image of only the anomaly.

major reasons. Along with the noise, the images contain edges and other complicated structures, leading to discontinuities and other singularities in the image intensity surfaces. Therefore, traditional methods for estimating smooth functions [e.g., Qiu (2009)] are usually inapplicable here. Additionally, a typical grayscale image contains a large number of pixels, leading to a high dimensional situation. In image based quality mon-

itoring, however, faulty regions in the image surface involve very small numbers of pixel coordinates. Such cases imply a *sparse* situation (Kang, 2022). Therefore, conventional multivariate control charts do not work well. As it is designed to detect *small* changes in the intensity function, it has potential applications in anomaly detection across various domains, including monitoring textured surfaces, steel and tile surfaces, and beyond. Figure 2.1 presents an example of a small change in the image of a textile material that is difficult to spot by visual inspection.

In this chapter, we propose an image monitoring control chart based on upper q quantile of pixelwise CUSUM-type statistics. The procedure is executed as follows: Firstly, to retain the edge information while denoising the observed image intensity function, we employ a state-of-the-art jump-preserving surface estimation technique instead of using a traditional nonparametric kernel regression estimate of the intensity function. Next, for each pixel, we construct a CUSUM-type statistic and sort them in ascending order. Finally, to detect anomaly in a small region, we use those CUSUM-type statistics which are larger than or equal to the upper q sample quantile of these statistics [Mei (2011)]. One of the major advantages of our method lies in its simplicity and flexibility. Based on the purpose of the applications, we can adjust the value of q to detect various amounts of anomalies. A small q is preferable for detecting anomalies or faults involving a small number of pixel coordinates, while a larger value of q is more suitable for larger fault regions.

This chapter makes the following contributions: (i) It introduces a CUSUM-type control chart designed specifically for monitoring "small" changes in image data. (ii) The proposed control chart can detect sparse anomalies in the image intensity function, and thus it expands the breadth of image monitoring techniques. (iii) One key advantage of the proposed algorithm is its flexible detection capability through the tuning parameter. Depending on the size of the anticipated anomaly or fault region, the proposed control chart can be made more suitable by tuning the value of upper quantile, and which enhances its practicality in various real applications. (iv) The chart can use information about the edges or jump location curves properly during the process of monitoring. (v) It demonstrates superior performance compared to various state-of-the-art benchmark methods.

2.2 Proposed Methodology

The proposed online monitoring of sequence of images consists of the following three steps. (i) Image pre-processing: to eliminate useless parts of the images, and to align them geometrically. (ii) Estimation of the image intensity functions: during sequential

monitoring, noisy image surface needs to be estimated using a nonparametric jump edge preserving state-of-the-art estimator. (iii) Online image monitoring: a Phase II procedure for online monitoring of images. The following subsections describe the aforementioned steps in a detailed manner.

2.2.1 Image pre-processing

Since we consider grayscale images only in this chapter, the first step of pre-processing is to transform RGB color images to grayscale images. Contrast enhancement, resizing, and standardization of image intensity value are also needed for further analysis. Moreover, geometric misalignment of the images of the same scene at different time points is common in image monitoring applications. In the image processing literature, the process of aligning those images geometrically is known as *image registration*. See Section 1.5 for more discussion regarding image registration problem. Throughout this chapter, however, we assume that the observed images from the production line are aligned properly.

2.2.2 Estimation of image intensity function

Under jump regression analysis (JRA) [Qiu (2005)], a 2-D monochrome image can be considered as a discontinuous regression surface where boundaries of the image objects are often considered as the point of discontinuities of the image intensity function or image intensity surface. In the JRA framework, the observed 2-D image intensities or image surface can be described by the following regression model:

$$w_{ij} = f(x_i, y_j) + \varepsilon_{ij}\ , \quad \text{for} \quad i, j = 1, 2, \ldots, n, \tag{2.2.1}$$

where $\{(x_i, y_j) = (i/n, j/n) : i, j = 1, 2, ..., n\}$ are equally spaced pixel coordinates in the design space $\Omega = [0, 1] \times [0, 1]$, f is the unknown true image intensity function, $N = n^2$ is the sample size and ε_{ij} are independent and identically distributed (i.i.d.) random noise with mean 0 and variance $\sigma^2 > 0$. Additionally, the unknown image intensity function f is assumed to be continuous except on the boundaries of the image objects. We assume that the image intensity function is piecewise continuous where the the number of pieces is finite. The locations of discontinuity points are usually curves. In the JRA literature, these curves are popularly known as jump location curves (JLCs) and we aim to preserve the JLCs while monitoring the images.

The main objective of the proposed image monitoring technique is to monitor the unknown discontinuous regression surfaces mentioned above. Hence, we need to accurately

estimate f first. Given the discontinuous nature of image surface, it is reasonable to employ an smoothing technique that can preserves the JLCs very well [Qiu (2009)]. In literature, there are several edge or jump preserving surface estimation techniques. In this chapter, we suggest a smoothing procedure proposed by Mukherjee and Qiu (2015), primarily based on the idea of local pixel clustering. This smoothing technique does not require any explicit edge detection, and the discontinuities and other complicated details of the images can be preserved well.

Now, to estimate f locally, we undertake the following three major steps. First, we determine whether a local neighborhood contains JLCs. Next, if JLCs are present, we partition the pixels in the local neighborhood into two groups based on their observed image intensity values. Finally, depending on whether the local neighborhood for a given pixel is located in the continuous or discontinuous region, we estimate the true intensity as follows: if the pixel lies in a continuous region, we approximate it by computing the local weighted average of all intensities within that neighborhood. Conversely, if the pixel is situated in a discontinuous region, then it is estimated by a weighted average of the observed intensities at pixels located in the same group as the given pixel. Mathematically, for a given pixel $(x, y) \in \Omega$, let us consider the circular neighborhood

$$B(x, y; h_n) = \{(u, v) : (u, v) \in \Omega, \sqrt{(u-x)^2 + (v-y)^2} \leq h_n\},$$

where $h_n > 0$ is a bandwidth parameter. After the pixels in $B(x, y; h_n)$ are clustered into two different groups $B_1(x, y; h_n)$ and $B_2(x, y; h_n)$, the true image intensity function at $f(x, y)$ can be approximated as the following:

$$\widehat{f}(x,y) = \begin{cases} \frac{\sum_{(x_i,y_j)\in B(x,y;h_n)} \phi_{ij} w_{ij}}{\sum_{(x_i,y_j)\in B(x,y;h_n)\phi_{ij}}}, & \text{if } B(x, y; h_n) \text{ does not contains JLCs} \\ \\ \frac{\sum_{(x_i,y_j)\in B_1(x,y;h_n)} \phi_{ij} w_{ij}}{\sum_{(x_i,y_j)\in B_1(x,y;h_n)\phi_{ij}}}, & \text{if } B(x, y; h_n) \text{ contains JLCs} \end{cases} \tag{2.2.2}$$

where ϕ_{ij} is the weight corresponding to the (i, j)-th pixel, and without any loss of generality, we assume that (i, j)-th pixel is located at group $B_1(x, y; h_n)$. See Mukherjee and Qiu (2015) for a rigorous discussion about the jump preserving estimator of the image intensity function.

2.2.3 Online monitoring in phase-II stage

In Phase-II stage, the k-th observed image surface can be expressed as a 2-D JRA model:

$$w_{ijk} = f_k(x_i, y_j) + \varepsilon_{ijk}; \quad \text{for} \quad i, j = 1, 2, \ldots, n \text{ and } \quad k = 1, 2, \ldots, \tag{2.2.3}$$

where w_{ijk} is the observed image intensity of the k-th image at (i,j)-th pixel, f_k is its true intensity function, and ε_{ijk} are i.i.d. random noise with zero mean and variance $\sigma^2 > 0$, i.e., similar to those as in (2.2.1). For detecting a step shift in image data, the change-point model for the (i,j)-th pixel can be expressed as:

$$w_{ijk} \sim \begin{cases} N\Big(f_0(x_i, y_j), \sigma^2\Big) & \text{if } k \leq \tau, \\ N\Big(f_1(x_i, y_j), \sigma^2\Big) & \text{if } k > \tau, \end{cases} \tag{2.2.4}$$

where τ is the change point. Note that, in many applications, the changes in the image involve relatively small number of pixels of the image. In literature, this phenomena is known as *sparsity*. In the SPC literature, there exists some control chart based on the sparsity assumption [Zou and Qiu (2009); Kang (2022)]. However, in this chapter, we propose an image monitoring scheme that is simple to interpret and also capable of handling the sparse nature of anomaly or change in the image intensity function. Therefore, to monitor a sequence of images of resolution $n \times n$, we propose a global charting statistic by combining the set of univariate CUSUM chart based on the sequence of each and every observed pixel coordinate $\{w_{ijk} : i, j = 1, 2, \ldots, n; \quad k = 1, 2, \ldots\}$. The intuition behind the proposed algorithm is as follows: Given a small number of pixels associated with the anomaly or fault regions, it is more effective to focus a few extreme values of the local pixelwise statistics rather than aggregating all of them together. At the k-th time point, the usual likelihood based local CUSUM statistic for monitoring the sequence of the intensity values at the (i,j)-th pixel is:

$$C_k(i,j) = \max\left[0, \quad C_{k-1} + \big(f_k(x_i, y_j) - f_0(x_i, y_j)\big)\left\{e_k(x_i, y_j) - \frac{1}{2}\big(f_k(x_i, y_j) - f_0(x_i, y_j)\big)\right\}\right], \tag{2.2.5}$$

where $e_k(x_i, y_j) = \big(w_{ijk} - f_0(x_i, y_j)\big)/\sigma$ follows a standard normal distribution. The likelihood based CUSUM chart raises a signal of intensity change at the (i,j)-th pixel if $C_k(i,j) > \zeta_{ij}$, where ζ_{ij} is an appropriately chosen threshold. Now, for the global charting statistic, let $C_k^{(1)} \leq C_k^{(2)} \leq \ldots \leq C_k^{(n^2)}$ be the increasing order statistics, where $C_k^{(\ell)}$ denotes the ℓ-th largest local CUSUM statistic. Then, our proposed monitoring scheme for online monitoring of the image data is as follows. We choose a value of r such as n^2, $\big[(1-q)n^2\big]$, etc. where q is an appropriately chosen value of the upper quantile depending on the anticipated size of the fault or anomaly region. At time k, the chart statistic raises an alarm at global level if

$$\sum_{\ell=r}^{n^2} C_k^{(\ell)}(i,j) > \Lambda_0, \tag{2.2.6}$$

where Λ_0 is chosen in such a way that the above global charting statistic can achieve the prefixed in-control average run length (ARL) value, called ARL_0.

Note that, for evaluating the local statistic (2.2.5) for Phase II monitoring, the true in-control image intensity function denoted as f_0, should be known. However, in reality, it is unknown and has to be estimated from the set of in-control images in the Phase I stage. Assume that there are m in-control images in Phase I. Then, based on the m observed samples, we define $\widehat{f}_0(x,y) = \frac{1}{m}\sum_{i=1}^{m}\widetilde{f}_i^{IC}(x,y)$, where $\widetilde{f}_i^{IC}$ is the local jump preserving smoothing estimate (2.2.2) of the i-th in-control image sample in Phase I. Moreover, to calculate the charting statistic, we have to use (2.2.2) to estimate f_k at each time point. Also, we suggest using the following estimator for σ:

$$\widehat{\sigma} = \sqrt{\frac{1}{n^2}\sum_{i,j}(w_{ij} - \widehat{f}_0(x_i, y_j))^2}. \tag{2.2.7}$$

Then, the proposed global charting statistic gives a signal if

$$\sum_{\ell=r}^{n^2}\widehat{C}_k^{(\ell)} > \lambda_0. \tag{2.2.8}$$

Note that, λ_0 is the estimated version of Λ_0. Next, we describe the algorithm to evaluate the value of λ_0 with a prefixed ARL_0 value.

Construction of the global control limit λ_0 with prefixed ARL_0 : In this chapter, λ_0 can be determined by a parametric bootstrap technique described below. Traditionally, to generate bootstrap sample of images we draw a re-sample from the residual images. However, in our example, residuals around the JLCs could be large, and as a result, there will be possibility of getting false discontinuity points in the image surface. To get rid from this problem, we suggest to generate k-th simulated image as $Z_{ijk}^{*} = \widehat{f}_0(x_i, y_j) +$ a random sample from $N(0, \widehat{\sigma}^2)$. Then, based on the bootstrap samples, global CUSUM charting statistic can be calculated by (2.2.8). For a given value of λ_0, we continue the above process until the global detection scheme in (2.2.8) raises a signal. Thus, a run is found and corresponding run length is determined. Repeat the above two steps B times to approximate the actual ARL_0 value by the sample mean of those B run lengths. Finally, if this value is smaller than the pre-determined ARL_0 value, then increase the previous λ_0 value, and otherwise decrease the previous λ_0 value. For searching the suitable value of λ_0, the bisection algorithm can be applied. For a related discussion regarding the searching algorithm, readers are referred to Qiu (2013).

Remark 2.2.1. *One major advantage of the proposed method is its inherent flexibility. This flexibility is evident in its construction, particularly the way it integrates both Max-*

CUSUM and Sum-CUSUM procedure to construct the global statistic based on top few local pixelwise CUSUM statistic. Max-CUSUM involves a CUSUM procedure based on the maximum of all local CUSUM statistics, i.e., when $r = n^2$, while Sum-CUSUM entails a global chart statistic based on the summation of all local CUSUM statistics i.e., $r = 1$. Intuitively, Max-CUSUM should work better when the fault region affects a very small number of pixel coordinates, whereas the Sum-CUSUM control chart is better otherwise. Consequently, a global CUSUM control chart based on top few local CUSUM statistics offers greater flexibility and practical relevance. However, the choice of r is crucial and it is subjective to prior knowledge. In this chapter, since the focus is on detecting very small changes we choose r to be n^2, $0.999n^2$, $0.995n^2$, and so forth.

2.3 Statistical Properties

In this section, we investigate a few important statistical properties regarding the proposed CUSUM chart for monitoring image data. Similar to the classical change point detection problem, our main objective is to provide a detection scheme that minimizes the detection delay subject to the pre-specified ARL_0. Based on the statistic in (2.2.6), the chart raises an alarm at the global level at time

$$T_{\text{sum}}^{(r)} = \inf\left\{k \geq 1 : \sum_{\ell=r}^{n^2} C_k^{(\ell)}(x, y) > \Lambda_0\right\},$$

where $\Lambda_0 > 0$ is the suitably chosen control limit so that the prefixed ARL_0 value is achieved. In statistical literature, $T_{\text{sum}}^{(r)}$ is known as the *stopping time* of the the proposed scheme. Then, the detection delay of the proposed scheme can be obtained by the following worst case detection delay defined in Lorden (1971):

$$\overline{E}^{(p_1,p_2,\ldots,p_s)}(T_{\text{sum}}^{(r)}) = \sup_{\tau=1,2,..} ess\sup E_\tau^{(p_1,p_2,\ldots,p_s)}\left[\max\left\{(T_{\text{sum}}^{(r)} - \tau + 1), 0\right\}\middle|\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_{(\tau-1)}\right].$$

Here, $\mathcal{I}_j$ indicates observed image intensity function at the j-th time point, and $E_\tau^{(p_1,p_2,\ldots,p_s)}$ is the expectation when at time τ, fault occurs by involving s many pixel coordinates. Note that $\{p_1, p_2, \ldots, p_s\}$ is the pixel coordinates associated with the fault region, and it is a subset of set of all pixel coordinates $\{(i/n, j/n) : i, j = 1, 2, ..., n\}$. We have the following propositions regarding the asymptotic optimality of the proposed monitoring scheme for detecting a change of fault in the image data. The first proposition is based on the average run length to false signal of change. The second proposition is regarding the detection delay of the proposed scheme.

Proposition 2.3.1. *In Phase II, for a sequence of $n \times n$ images, as $\Lambda_0 \to \infty$,*

$$E_\infty\left\{T^{(r)}_{sum}(\Lambda_0)\right\} \geq \Big(1 + o(1)\Big)\left[\frac{\exp(\Lambda_0)}{1 + \Lambda_0 + \Lambda_0^2/2! + \ldots + \Lambda_0^{(n^2-1)}/(n^2-1)!}\right],$$

where E_∞ is the expectation when there is no change occur in the process.

The above proposition establishes a lower bound of the run length when there is no change in the process. It indicates low false alarm probability of raising signal when there is no change in the images. Our next proposition is based on detection delay when there is a sustained change in the images.

Proposition 2.3.2. *For any possible subset $\mathcal{S} = \{p_1, p_2, \ldots, p_s\}$ of pixel locations $\{(i/n, j/n) : i, j = 1, 2, ..., n\}$, as $\Lambda_0 \to \infty$,*

$$\overline{E}^{(p_1,p_2,\ldots,p_s)}(T^{(r)}_{sum}(\Lambda_0)) \leq \frac{2\Lambda_0\left(\frac{n^2}{n^2-r+1}\right)\sigma^2}{\sum_{(x,y)\in\mathcal{S}}(f_0(x,y) - f_1(x,y))^2} + O(1).$$

Proofs of these propositions using Theorem 1 of Mei (2010) are provided in Appendix A.1.

2.4 Numerical Studies

In this section, we present numerical examples concerning the performance of the proposed method in comparison with a number of state-of-the-art methods. We conduct the comparative studies on a set of simulated test images and a real image. To show the effectiveness of the proposed method, we present the mean and standard deviation of the out-of-control ARL, denoted as ARL_1, as a performance evaluation metric. As the proposed method is a CUSUM procedure, the control chart statistic depends on the previous images, therefore steady state calculation of ARL_1 is a reasonable choice. Additionally, since there is only one image at each time point, there is no requirement for temporal aggregation [Zwetsloot and Woodall (2021)]. Throughout this section, we intentionally corrupt the images with additive Gaussian noise. We hereby consider the following competing methods:

- Wavelet-based image surveillance by Koosha et al. (2017)
- Supervised-learning based approach by Bui and Apley (2018a)

Note that the aforementioned methods share similar assumptions with the proposed method and find applications in analyzing industrial images.

2.4.1 Brief description of the competing methods

Koosha et al. (2017) proposes a nonparametric profile monitoring approach where each image is decomposed into a set of 1D profiles and monitors them using a GLR (generalized likelihood ratio) control chart. To capture frequency domain characteristics, it utilizes wavelet transformation and tracks these features over time. In this scenario, every row of a gray-scale image of size $n \times n$ is treated as 1D profile. If there are d estimated features from each profile, then each image feature becomes a vector of dimensions $nd \times 1$. Let's denote the coefficient vector of an image as $\mathbf{\Gamma}^{\intercal} = (\gamma_{11}, \ldots, \gamma_{1d}, \ldots, \gamma_{n1}, \ldots, \gamma_{nd})$. To monitor the images in Phase-II, the GLR control chart statistic for the k-th Phase II image, as defined by Koosha et al. (2017), is denoted as:

$$R_k = \max_{\tau, l \in \Gamma} \frac{(k-\tau)}{\sigma_l^2} \Big(\widehat{\gamma}_{1,\tau,k}(l) - \widehat{\gamma}_0(l) \Big)^2, \quad k = 1, 2, \ldots. \tag{2.4.1}$$

Here, $\widehat{\gamma}_{1,\tau,s}(l) = (s-\tau)^{-1} \sum_{t=\tau+1}^{s} \widehat{\gamma}_{t,l}$, where $\widehat{\gamma}_{t,l}$ is the l-th element of $\mathbf{\Gamma}^{nd \times 1}$ for the t-th sample, $\gamma_0(l)$ is the l-th element of $\mathbf{\Gamma}^{nd \times 1}$ corresponding to the nominal image, and σ_l^2 denotes the variance of the l-th coefficient computed from Phase I samples.

Bui and Apley (2018a) propose a supervised-learning based approach for monitoring and diagnosing texture surface related anomalies in manufactured products. The central idea is monitoring the behaviour of local residuals over time. To measure the degree of deviation from the in-control behaviour, they propose a general spatial moving statistic (SMS) based on one sample A-D (Anderson and Darling) statistic [Anderson and Darling (1954)]. For online monitoring, chart statistic for k-th Phase II images can be expressed as:

$$S_k = \max_{(i,j)} \text{SMS}_{k,ij}, \tag{2.4.2}$$

where $\text{SMS}_{k,ij}$ is the spatial moving statistic at the (i,j)-th pixel at k-th time point. Readers are referred to Bui and Apley (2018a) for more details about this algorithm. Note that this is a Shewhart type control chart. For implementing this procedure we use *spc4sts* [Bui and Apley (2021)] package from CRAN-R.

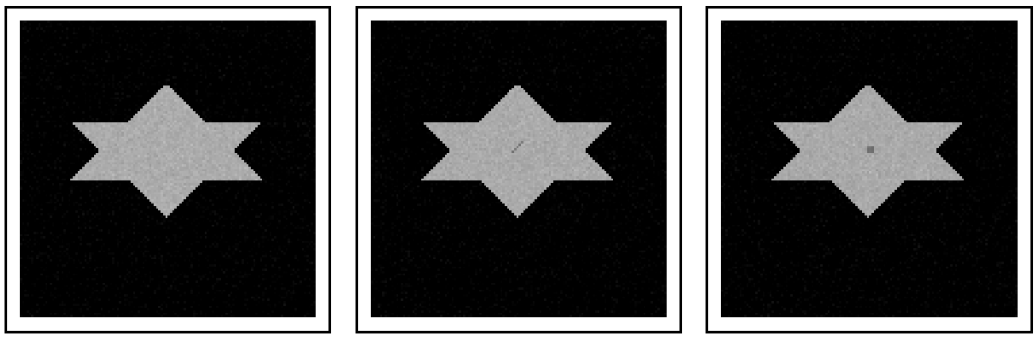


Figure 2.2: Images from left: (a) in-control simulated image, (b) out-of-control image with a scratch in the middle, and (c) out-of-control image with a square spot in the middle.

2.4.2 Simulations

In our simulation study, the true in-control image intensity function described in (2.2.4) has the following expression:

$$f_0(x, y) = \mathbb{I}[(x, y) \in S], \quad (x, y) \in [0, 1] \times [0, 1], \tag{2.4.3}$$

$\mathbb{I}(.)$ is the indicator function. See extreme left of Figure 2.2 for the simulated image. The set of pixels into the star is denoted as S. Consequently, to generate Phase I and phase II samples, we add point-wise Gaussian noise generated from $N(0, 0.03^2)$. We carry out the simulation study under the following set-ups: (i) we consider two different image resolutions 64×64 and 128×128, and (ii) to generate out-of-control images, we introduce two types of changes or faults: scratch and square spot on the true or nominal image intensity function f_0. Moreover, for each type of fault, we consider three different degrees of intensity change: 0.1, 0.2, 0.3, and 0.5. Examples of in-control and out-of-control images are shown in Figure 2.2. The faulty region are located around the centre of the star. In this study, based on 500 independent replications, we fix the in-control ARL, i.e., ARL_0 at 20. It is important to highlight that the pre-defined ARL_0 value is relatively smaller as compared to the values in traditional process monitoring literature. $ARL_0 = 20$ is a reasonable choice due to limited availability of in-control sample images in Phase I stage. This is often the situation in most practical applications, as collection of images involve additional cost to the concerned authority. To set the in-control limit with a prefixed ARL_0, we follow the description in Section 2.2.3. For performance evaluation of the proposed method, we perform a comparative numerical analysis based on the sample mean and standard deviation of ARL_1. Our simulation results are summarized in Tables 2.1 and 2.2. For comparison purpose, we consider two different versions of the proposed algorithm. One is based on the maximum CUSUM statistic, denoted as Max-CUSUM, and others are based on the upper q quantile of the CUSUM statistics, where $q = 0.5\%$, $q = 5\%$, and $q = 10\%$. Based on the definition of ARL, we anticipate small value of

ARL_1 for various out-of-control images, if the changes are detected by the concerned methods. In Phase II monitoring, we set $\tau = 4$. That is for each replication of run length, a sustained shift occurs on the fifth image. If the chart statistic raise signal before the 5-th time point, we exclude that run from the calculation of ARL_1.

		Proposed Algorithm				Competing Methods	
Fault type	Shift Size	Max-CUSUM	Upper-q-CUSUM ($q = 0.5\%$)	Upper-q-CUSUM ($q = 5\%$)	Upper-q-CUSUM ($q = 10\%$)	Bui & Apley Bui and Apley (2018a)	Koosha et a;. Koosha et al. (2017)
Scratch	0.5	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	18.37 (20.97)	2.02 (0.78)
	0.3	1.01 (0.07)	1.01 (0.07)	1.01 (0.10)	1.01 (0.09)	18.14 (20.83)	4.06 (1.90)
	0.2	1.67 (0.99)	1.70 (1.00)	1.96 (1.28)	2.33 (1.68)	19.27 (20.61)	6.83 (3.95)
	0.1	17.94 (17.24)	18.23 (17.17)	17.26 (17.80)	25.00 (26.63)	17.80 (20.91)	13.62 (10.00)
Square	0.5	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	18.89 (22.08)	1.07 (0.26)
	0.3	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	12.34 (15.04)	2.22(0.78)
	0.2	1.26 (0.53)	1.26 (0.54)	1.31 (0.62)	1.41 (0.71)	18.25 (19.32)	4.21 (1.91)
	0.1	16.86 (16.43)	15.91 (15.79)	13.24 (12.72)	19.43 (20.10)	16.37 (19.33)	13.33 (8.50)

Table 2.1: Phase II performance comparison based on images with resolution 64×64. Each entry presents the sample mean of ARL_1 values along with the sample standard deviation within parenthesis, based on 500 independent replications.

		Proposed Algorithm				Competing Methods	
Fault type	Shift Size	Max-CUSUM	Upper-q-CUSUM ($q = 0.5\%$)	Upper-q-CUSUM ($q = 5\%$)	Upper-q-CUSUM ($q = 10\%$)	Bui & Apley Bui and Apley (2018a)	Koosha et a;. Koosha et al. (2017)
Scratch	0.5	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	28.37 (32.86)	1.96 (0.62)
	0.3	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	29.38 (33.42)	4.45 (1.53)
	0.2	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	29.81 (32.70)	8.56 (3.39)
	0.1	1.00 (0.00)	1.54 (0.84)	1.56 (0.93)	1.92 (1.25)	30.71 (34.62)	18.30 (9.75)
Square	0.5	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	26.58 (30.88)	1.03 (0.17)
	0.3	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	28.78 (33.09)	2.08 (0.63)
	0.2	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	29.16 (31.09)	3.69 (1.22)
	0.1	1.01 (0.10)	1.01 (0.11)	1.02 (0.15)	1.05 (0.23)	29.68 (32.72)	9.50 (4.21)

Table 2.2: Phase II performance comparison based on images with resolution 128×128. Each entry presents the sample mean of ARL_1 values along with the sample standard deviation within parenthesis, based on 500 independent replications.

Based on the summary provided in Tables 2.1 and 2.2, the following observations can be made regarding the proposed method: (i) As the magnitude of the change or shift increases, the average run length of detecting a rising signal decreases. This trend is consistent with our expectations and intuition. (ii) In the current simulation study, the performance of Max-CUSUM and Upper-q-CUSUM methods are comparable. For smaller shift sizes, Max-CUSUM exhibits slightly better performance, which aligns with our intuitive expectations. Regarding the competing methods, it is quite evident that the method proposed by Bui and Apley (2018a) fails to detect the change in all situations of the current simulation study. This failure is attributed to the fact that it is mainly designed to detect larger shifts in the pattern of a textured surface. Since it relies on a Shewart-type control chart, it is more suitable for identifying large transient shifts rather than smaller persistent shifts. On the other hand, the method proposed by Koosha et al. (2017) performs relatively better. However, the comparison of the proposed algorithm is quite illustrative. It outperforms the existing state-of-the-art methods in almost all

situations. In particular, for the image of resolution 64×64 with shift size 0.1, both of them fail to detect. However, for 128, the proposed method works perfectly. The failure of the proposed method in the case of 64×64 images can be attributed to relatively poorer performance of image denoising in low resolution images. Except for that specific setting, our proposed algorithm is the clear winner for detecting the change nearly perfectly.

Additional Simulation Studies: To emphasize the robustness and practical applicability of the proposed method, we hereby perform a comparative analysis with more types of simulated texture images. The true texture surface has the following functional form:

$$f_0(x, y) = \begin{cases} \frac{1}{4}\sin(128(x+y)) & \text{if } 0.625 \leq x + y \leq 1.375 \\ \frac{1}{4}\sin(32(x+y)) & \text{otherwise.} \end{cases}$$

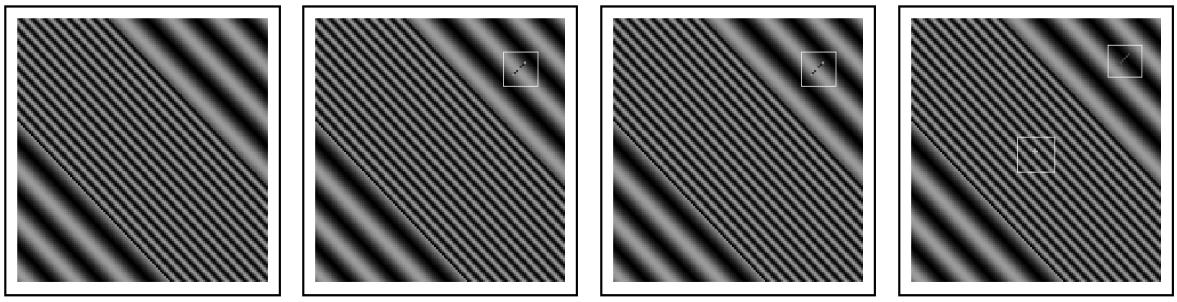


Figure 2.3: Images from left: (a) in-control simulated image, (b) out-of-control image: scratch, (c) out-of-control image: pattern change, and (d) out-of-control image: multiple faults.

Examples of simulated these texture images are shown in Figure 2.3. The left panel of Figure 2.3 shows the in-control image, whereas the rest show various out-of-control images. Faulty regions are highlighted by rectangular boxes. To simulate the out-of-control images, we consider three different cases: (i) a small scratch on the surface, (ii) a change in the texture pattern, and (iii) multiple fault regions. Note that, we carry out the simulation study with a similar set-up as before. A comparative analysis of the simulation study is summarized in Table 2.3 when the image resolutions are 128×128. It is evident that the proposed Upper-q-quantile based method outperforms its competitors nearly in all situations. Additionally, its performance aligns well with our expectations from earlier simulations. Since the changes are small, the detection power increases as the value of q decreases in the Upper-q-CUSUM control chart. Among the competing methods, the wavelet-based approach by Koosha et al. (2017) performs relatively better, while the method by Bui and Apley (2018a) shows strong performance for detecting pattern changes in the texture surface. However, this method is not recommended for detecting small changes. Thus, in the given scenarios, the proposed chart statistic is a clear winner and its power of raising a change signal in presence of very small change(s)

in the image intensity function demonstrates its reliability and practical applicability in various scenarios.

	Proposed Algorithm				Competing Methods	
Fault Type	Max-CUSUM	Upper-q-CUSUM ($q = 0.5\%$)	Upper-q-CUSUM ($q = 5\%$)	Upper-q-CUSUM ($q = 10\%$)	Bui & Apley Bui and Apley (2018a)	Koosha et al. Koosha et al. (2017)
Scratch	1.00 (0.00)	1.00 (0.00)	1.33 (0.65)	1.78 (1.31)	13.29 (12.61)	3.27 (1.84)
Pattern change	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.56 (0.74)	1.00 (0.00)
Multiple faults	1.00 (0.00)	1.05 (0.21)	1.49 (0.94)	1.87 (1.42)	15.73 (16.12)	2.61 (1.20)

Table 2.3: Phase II performance comparison on simulated textured images with resolution 128×128. Each entry presents the sample mean of ARL_1 values along with the sample standard deviation within parenthesis, based on 500 independent replications.

2.5 A Real Image Example

In this section, we illustrate the performance of the proposed control chart on real images of textile materials. Example images of in-control and out-of-control textile materials are shown in Figure 2.4. Note that, the fabric pattern of the textile image are very complicated and it is challenging to identify small anomalies from the observed images. Textile image data are available in the package "*textile*" from CRAN-R [Bui and Apley (2018b)]. We induce the ellipsoidal anomalies using **imposedefect( )** function from the package "*spc4sts*" [Bui and Apley (2021)].

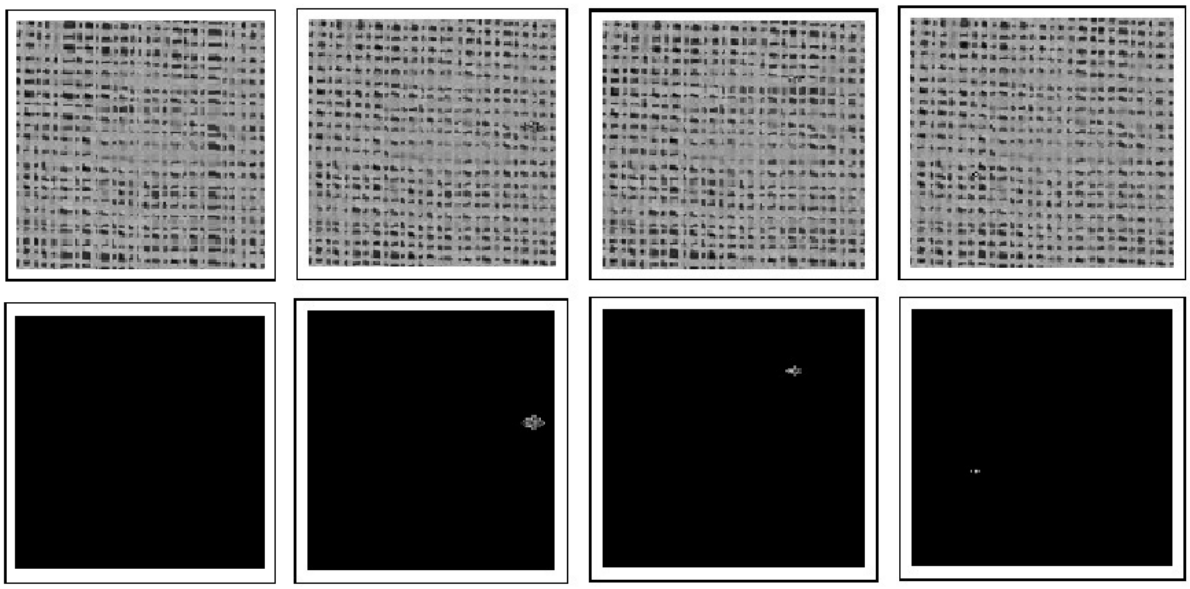


Figure 2.4: Top row: In-control and out-of-control real textile images. The first image is an in-control image. Remaining three are different out-of-control images with ellipsoidal fault regions. Bottom row: Images indicating the locations of the fault regions. The first image is completely black because there is no fault region.

In this study, we consider three distinct out-of-control images based on the size of the ellipsoidal fault (See Figure 2.4). Initially, as a part of pre-processing, we resize the image resolution to 128×128 image and normalize the intensities so that their values lie between 0 and 1. Subsequently, to generate Phase I and Phase II samples, we add point-wise Gaussian noise with mean zero and standard deviation 0.01. In this example, we fix the in-control ARL, i.e., ARL_0 at 20, and compute the control limit based on 500 independent replications. Since our target is to detect subtle changes in the textile surface, we choose $q = 0.001$, i.e., we construct the chart statistic based on top 0.1% pixelwise CUSUM statistics. Figure 2.5 presents the performance of the proposed control chart. In Phase II stage, we consider 10 images of the texture surface. Among these 10 images, the first 4 are in-control, and a sustained shift occurs on the 5-th image. Thus, in the given situation, the change-point $\tau = 5$. From Figure 2.5, we see that the proposed method performs very well in all the three situations mentioned above. It is evident that the chart statistic can detect the change at the 5-th time-point itself in each case.

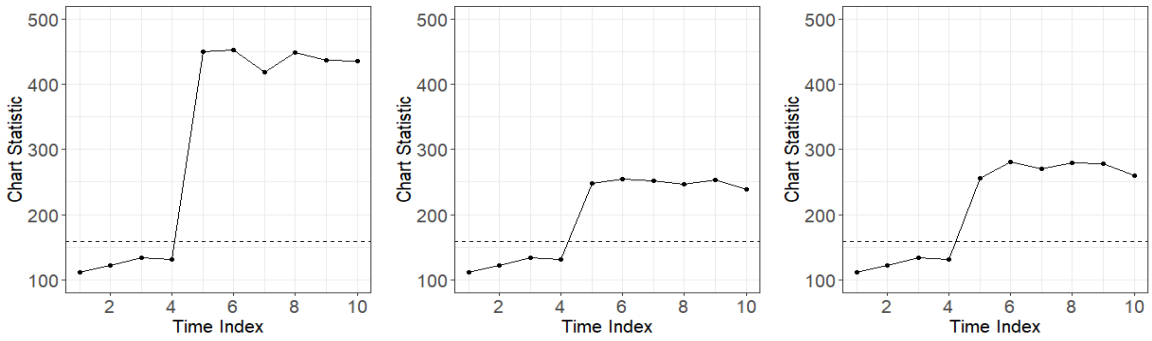


Figure 2.5: Performance of the upper 0.1% CUSUM control chart in Phase II corresponding to the three changes described in Figure 2.4.

2.6 Concluding Remarks

In this chapter, we propose an efficient image monitoring algorithm for detecting small changes from the noisy images. Numerical examples ensure superiority of the proposed method to its competitors in various situations. There are several directions for extending our current research on image monitoring. The proposed method assumes the conditions of jump regression analysis, it also assumes point-wise independent Gaussian noise. An immediate extension of the proposed method is to monitor sequence of images in presence of spatio-temporally correlated noise.

Additionally, our current method assumes that images are geometrically aligned properly. A method that can handle geometric misalignment by its construction is always preferable. Performance-wise superiority of our proposed method is undeniable. How-

ever, our current method focuses on images with moderate resolutions. Monitoring high-resolution images might be computationally demanding for our proposed control chart. Much future research is needed to address all these aspects.

Chapter 3

Monitoring Images Using Jump Location Curves[†]

3.1 Introduction

As because images have been traditionally used as a form of data in the chemical and manufacturing processes, there exists a number of SPC methods for monitoring images. However, none of them uses detected edges for image monitoring. The novelty of our approach lies in the fact that it explicitly uses detected edges, one of the most important features of an image. In literature, almost all existing methods are based on image intensity values, and therefore, they are too sensitive for a very small change in noise level and for a insignificant alternation of image intensity values. In that sense our method is more robust in presence of such insignificant changes. To our best knowledge, there is no existing study on monitoring image data considering edges as main feature for image surveillance. Under JRA [Qiu (2005)], a 2-D grayscale image intensity function can be considered as a discontinuous regression surface, where the location of the discontinuities are often in the boundaries of the image object. The locations of discontinuity points are usually curves, known as jump location curves (JLC). For more details, readers are referred to Section 1.2. The main purpose of this chapter is to propose a Shewhart-type [Shewhart (1931)] control chart based on JLCs for monitoring images. Essentially, we aim to monitor the JLCs in the regression surface over time. For this purpose, we first detect the edge pixels from the images using an appropriate local smoothing procedure and then for comparison we choose a quantitative similarity measure. In literature, commonly used similarity measures are mean square deviation (MSD) between two images or the Pearson

[†]This chapter is based on the publication Roy and Mukherjee (2024a): **Roy, A.** and Mukherjee, P.S. "A control chart for monitoring images using jump location curves", *Quality Engineering*, 2024, 36(2), 439–452. DOI: 10.1080/08982112.2023.2232441.

correlation among the intensities of two image intensity matrices. However, the major drawback of these measures are: they can not take into account the spatial structures and other complicated details of an image. In our proposed methodology, we choose the *Hausdorff* distance between the point-sets of detected edge pixels of two images, and thus all issues mentioned above can be addressed properly.

3.2 Edge Detection by LLK Smoothing

As discussed earlier, a very important step in our proposed image monitoring control chart is to identify the edge pixels of the images using a reasonable edge detector. Any suitable edge detector can be used for this purpose. Many existing edge detection methods are based on first-order derivatives, [e.g., Canny (1986), Qiu and Bhandarkar (1996)] or second-order derivatives [e.g., Sun and Qiu (2007), Joo and Qiu (2009), Torre and Poggio (1986), Clark (1988)] of the image intensity function. in this chapter, we use edge detection method based on local linear kernel (LLK) smoothing.

Literature on jump regression analysis considers estimating jump location curves (JLCs) in the design space. However, presence of multiple JLCs or the JLCs crossing over in the design space make it very difficult to describe the JLCs mathematically due to the global nature of the curves [Qiu (2005)]. In the digital image setup the more flexible and appropriate method of jump detection is to consider JLCs as a point set in the design space. Qiu (1998) and Qiu and Yandell (1997) estimate JLCs as point set by a local linear estimator. We consider a similar approach using a kernel function in the circular neighbourhood and it is known as jump detection by local linear kernel (LLK). Under the jump regression analysis framework, a 2-D image can be expressed by the following regression model:

$$w_{ij} = f(x_i, y_j) + \varepsilon_{ij}\ , \quad \text{for} \quad i, j = 1, 2, \ldots, n, \tag{3.2.1}$$

where $\{(x_i, y_j) : i, j = 1, 2, ..., n\}$ are equally spaced pixel coordinates in the design space $\Omega = [0, 1] \times [0, 1]$, f is the unknown image intensity function, $N = n^2$ is the sample size and ε_{ij} are independent and identically distributed (i.i.d.) random errors with mean 0 and variance $\sigma^2 > 0$. Now, to detect edge pixels, we consider a circular neighborhood at each pixel (x, y) as $B(x, y, h_n) = \{(u, v) : \sqrt{(u - x)^2 + (v - y)^2} \leq h_n\}$ where $h_n > 0$ is a bandwidth parameter. Then LLK smoothing procedure is accomplished by the following optimization problem:

$$\min_{a,b,c} \sum_{(x_i,y_j)\in B(x,y,h_n)} \{w_{ij} - (a + b(x_i - x) + c(y_j - y)\}^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right), \tag{3.2.2}$$

where K is a radially symmetric bivariate density kernel with support $\{(x, y) : x^2 + y^2 \leq 1\}$. The estimate of (a, b, c) from the above minimization problem (3.2.2) is denoted as $(\hat{a}, \hat{b}, \hat{c})$. When there is no jump, $\hat{a}(x, y)$ is the LLK estimate of $f(x, y)$, and $(\hat{b}(x, y), \hat{c}(x, y))'$ is the LLK estimator of the gradient vector $\boldsymbol{\beta(x, y)} = (f_x(x, y), f_y(x, y))'$. One can easily verify that the estimator has the following expression:

$$\begin{pmatrix} \widehat{a}(x,y) \\ \widehat{b}(x,y) \\ \widehat{c}(x,y) \end{pmatrix} = \begin{bmatrix} m_{00} & m_{10} & m_{01} \\ m_{10} & m_{20} & m_{11} \\ m_{01} & m_{11} & m_{02} \end{bmatrix}^{-1} \begin{bmatrix} \sum w_{ij} K_{ij} \\ \sum w_{ij}(x_i - x) K_{ij} \\ \sum w_{ij}(y_j - y) K_{ij} \end{bmatrix} \tag{3.2.3}$$

Where $m_{pq} = \sum (x_i - x)^p (y_j - y)^q K_{ij}$ for $p, q = 0, 1, 2$, and $K_{ij} = K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)$. The estimated gradient vector at (x, y) is given by $\widehat{\boldsymbol{\beta}}(x, y) = (\widehat{b}(x, y), \widehat{c}(x, y))'$ which carries information about jump at point (x, y). Larger value of it indicates that underlying regression function f would be steeper around the point (x, y) and (x, y) is a possible jump point or an edge pixel. However, when f is steep but continuous in the circular neighborhood $B(x, y, h_n)$, the value of $\widehat{\boldsymbol{\beta}}(x, y)$ would still be large. Therefore, large value of $\widehat{\boldsymbol{\beta}}(x, y)$ does not guarantee that (x, y) is a jump point. To get rid from the slope effect we consider two neighboring points (x_{P_1}, y_{P_1}) and (x_{P_2}, y_{P_2}) such that:

- distance of (x_{P_1}, y_{P_1}) and (x_{P_2}, y_{P_2}) from (x, y) is $2h_n$, i.e., there is no overlap among the circular neighborhoods of the three points (x_{P_1}, y_{P_1}), (x, y), and (x_{P_2}, y_{P_2}).
- The neighboring points are along the direction of $\widehat{\boldsymbol{\beta}}(x, y)$, i.e., the estimated gradient direction at (x, y).

Using these, the jump detection criteria is defined as the following:

$$\lambda(x, y) = \min \left\{ ||\widehat{\boldsymbol{\beta}}(x, y) - \widehat{\boldsymbol{\beta}}_{P_1}(x, y)||, ||\widehat{\boldsymbol{\beta}}(x, y) - \widehat{\boldsymbol{\beta}}_{P_2}(x, y)|| \right\},$$

where $\widehat{\boldsymbol{\beta}}_{P_1}(x, y)$ and $\widehat{\boldsymbol{\beta}}_{P_2}(x, y)$ are the estimated gradient vectors at the neighboring points (x_{P_1}, y_{P_1}) and (x_{P_2}, y_{P_2}), respectively and $||.||$ is the Euclidean norm. We decide that there is no jump at the point (x, y) if the value of $\lambda(x, y)$ is relatively small. Therefore (x, y) is on the JLC or it is a detected jump point if

$$\lambda(x, y) > v_n$$

where v_n is a threshold parameter.

3.3 Phase-I Monitoring of Image Data

3.3.1 Model description

in this chapter, we mainly consider 2-D grayscale images. A sequence of m images of the dimension $n \times n$ each can be expressed by the following 2-D JRA model:

$$\zeta_{ijk} = f_k(x_i, y_j) + \varepsilon_{ijk}; \quad \text{for} \quad i, j = 1, 2, ...n \quad \text{and} \quad k = 1, 2, ...m. \tag{3.3.1}$$

where (x_i, y_j) is the (i, j)-th pixel coordinate of the k-th image. ζ_{ijk} is the observed intensity and $f_k(x_i, y_j)$ is the true intensity value of the k-th image at the pixel (x_i, y_j). ε_{ijk}'s are random errors with mean 0 and variance $\sigma^2 > 0$. Further, we assume that images are independent with each other and they are geometrically aligned. In case they are not geometrically aligned, they have to be registered appropriately.

3.3.2 Edge detection and Hausdorff distance

In Section 3.2, we describe a method to detect the JLC as a point set. We assume that in Phase-I, we have images of batch size one from m different time points and our primary goal is to detect if there is any change in the image objects. In Phase-I, our main objective is to construct a chart statistic and estimate the corresponding parameters and modify them simultaneously with data of different time points.

Let $\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_m$ be the m sample images with dimensions $n \times n$ from an image process at m different time points. We assume that the images contain noise of various levels. We define that estimated true in-control image as $\widehat{\mathcal{I}}_{IC} = \frac{1}{m}\sum_1^m \mathcal{I}_i$. Now, by LLK smoothing as described previously, the estimated point set of edge pixels of sample images is denoted as $\{\widehat{E}_k : k = 1, 2, \ldots, m\}$. The estimated edge pixel of the estimated true image $\widehat{\mathcal{I}}_{IC}$) is denoted as $\widehat{E_0}$. Since we are considering edge pixels as a point set, a very well known measure to calculate the dissimilarity between two point sets is the *Hausdorff distance* measure. In image processing literature, it is very popular with lots of applications in image matching, object tracking , image comparison etc. It measures how close two subsets from the same metric space are from each other. Being a distance measure on some metric space it has the following properties:

- The *Hausdorff* distance between two sets is bounded if the sets are bounded.
- The *Hausdorff* distance between two sets having the same closure is zero.
- Triangle inequality holds for *Hausdorff* distance.

In our context, images are defined on the compact design space $[0,1] \times [0,1]$ with equally spaced discrete pixel coordinates. Hence the distance is bounded and smaller the value of the *Hausdorff* distance between the point sets of detected edge pixels of two images more similar the images are. Therefore, for point sets of edge pixels $\widehat{E_i}$ and $\widehat{E_j}$ the *Hausdorff* distance is denoted as $D_H(\widehat{E_i}, \widehat{E_j})$ and expressed in Qiu (2005) by the following:

$$D_H(\widehat{E_i}, \widehat{E_j}) = \max \left\{ \sup_{(p,q)\in\widehat{E_i}} \inf_{(p',q')\in\widehat{E_j}} ||(p,q)-(p',q')|| \ , \ \sup_{(p,q)\in\widehat{E_j}} \inf_{(p',q')\in\widehat{E_i}} ||(p,q)-(p',q')|| \right\} \tag{3.3.2}$$

3.3.3 Construction of control chart for image monitoring

Now we construct an edge based control chart based on the sample images of size m. As mentioned earlier, most of image monitoring methods in the literature are intensity based and a vast majority of them uses GLR chart or MGLR chart to construct control chart. in this chapter, we are trying to form an edge based control chart to monitor an image process over time. To construct the control chart, we have to first define the chart statistic that could help us construct upper control limit (UCL) for the process and also help practitioners to detect changes in online monitoring stage. From the previous section, we know that the estimated point set of the edge pixels of an in-control (IC) image is defined as $\widehat{E}_0$. $\{\widehat{E}_i : i = 1, 2, \ldots, m\}$ are the corresponding estimated point sets of edge pixel of the sample images. A natural statistic can be defined as:

$$T = \frac{1}{m} \sum_{k=1}^{m} D_H(\widehat{E}_k, \widehat{E}_0).$$

Then, the chart will give signal if $T > t^*$, where $t^* > 0$ is a control limit. The value of t^* can be obtained from the $(1-\alpha)$-quantile of empirical distribution of T where α is given nominal IC false alarm rate (FAR). To get empirical distribution of the chart statistic bootstrap technique can be used.

3.4 Phase-II Monitoring of Image Data

Let us denote the true in-control (IC) image as $\mathcal{I}_{IC}$. In reality, it is usually unknown and has to be estimated based on observed Phase-I image sample $\{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_m\}$ of size m. As discussed before, estimated true image is defined as $\widehat{\mathcal{I}}_{IC} = \frac{1}{m}\sum_1^m \mathcal{I}_i$, and the point set of detected edge pixels of this image is denoted as $\widehat{E}_0$. In Phase-II step, ℓ-th observed

image can be expressed as 2-D JRA model as:

$$w_{ij\ell} = f_\ell(x_{i\ell}, y_{j\ell}) + \epsilon_{ij\ell} \quad \text{for} \quad i, j = 1, 2, \ldots, n \text{ and } \quad \ell = 1, 2, \ldots, \tag{3.4.1}$$

where $w_{ij\ell}$ is the observed image intensity and the other related variables can be described similarly to those in the model (3.3.1). Image registration is also a necessary step in Phase-II monitoring as well. To register the images, we use any IC image as a baseline image. There will not be substantial variations if we use different IC images as baseline image. After registration, we estimate the point set corresponding to the edge pixels of the images in Phase-II using LLK method and those are denoted as $\{\widehat{E}_\ell : \ell = 1, 2, \ldots\}$. Then, the *Hausdorff* distance of the edge point set of ℓ-th image in Phase-II with the edge point set of mean image is calculated by $T_\ell = D_{H,\ell}(\widehat{E}_\ell, \widehat{E}_0)$. The chart will give a signal of change if $T_\ell > t_0$, where t_0 has been chosen in such a way that the chart statistic can reach the prefixed ARL_0 value.

Construction of the control limit with a pre-fixed ARL_0: Without any loss of generality, we assume that the first IC image as the reference image. We apply local piece-wise smoothing procedure for image surface estimation and calculate the residuals $\{\widehat{\varepsilon}^*\}$ and $\widehat{\sigma}$ from the reference image. Traditionally, to generate bootstrapped images, we draw a re-sample of size n^2 from the set of residuals $\{\widehat{\varepsilon}^*(x_i, y_j) : i, j = 1, 2 \ldots n\}$. However, in our e example, the residuals around the estimated JLCs could be large, and as a result, there will be a few falsely detected edge pixels in the image at the time of edge detection at the bootstrapped images. To overcome this problem, we generate the ℓ-th simulated image by adding Gaussian noise $N(0, \widehat{\sigma}^2)$ to each pixel, and calculate the statistic T_ℓ based on this generated bootstrapped sample image. For a given value of t_0, we continue the above process until we get a signal of change. Thus, one run is found and the corresponding run length is calculated. Finally, we repeat the above two steps B times and average those B run lengths to estimate the actual value of the ARL_0. If the estimated value based on B replications is smaller than the pre-fixed ARL_0, then we should increase the specified value of t_0 and if the estimated value of the ARL_0 is larger than the pre-fixed ARL_0, then we should decrease the specified value of t_0.

3.5 Selection of Procedure Parameters

In the proposed image monitoring procedure, the parameter v_n in Section 3.2 should be chosen carefully to get a good performance in edge detection.

If no jump exists in $B(x, y, h_n) \cup B(x_{P_1}, y_{P_1}, h_n) \cup B(x_{P_2}, y_{P_2}, h_n)$, for a given constant $v_n > 0$,

$$
\begin{aligned}
P[\lambda(x,y) > v_n] &\leq P[||\widehat{\beta}(x,y) - \widehat{\beta}_{P_1}(x,y)|| > v_n] \\
&= P[(\widehat{b}(x,y) - \widehat{b}_{P_1}(x,y))^2 + (\widehat{c}(x,y) - \widehat{c}_{P_1}(x,y))^2 > v_n^2] \\
&= E\{P[(\widehat{b}(x,y) - \widehat{b}_{P_1}(x,y))^2 + (\widehat{c}(x,y) - \widehat{c}_{P_1}(x,y))^2 > v_n^2 | \widehat{b}(x,y), \widehat{c}(x,y)]\}.
\end{aligned}
$$

For fixed $\widehat{b}(x,y)$ and $\widehat{c}(x,y)$, $(\widehat{b}(x,y) - \widehat{b}_{P_1}(x,y))^2 + (\widehat{c}(x,y) - \widehat{c}_{P_1}(x,y))^2/\sigma_{P_1}^2$ approximately follows χ_2^2 distribution, where $\sigma_{P_1}^2 = Var(\widehat{b}_{P_1}(x,y))$. Now from expression (3.2.1), we have

$$
\sigma_{P_1}^2 = \sigma^2 \frac{\sum (x_i - x_{P_1})^2 K_{P_1}^2}{\{\sum (x_i - x_{P_1}) K_{P_1}\}^2},
$$

where $K_{P_1} = K(\frac{x_i - x_{P_1}}{h_n}, \frac{y_i - y_{P_1}}{h_n})$. Therefore, a natural choice for v_n is

$$
v_n = \widehat{\sigma} \sqrt{\frac{\chi_{2,1-\alpha_n}^2 \sum (x_i - x_{P_1})^2 K_{P_1}^2}{\{\sum (x_i - x_{P_1}) K_{P_1}\}^2}},
$$

where $\chi_{2,1-\alpha_n}^2$ is the $(1-\alpha_n)$ quantile of χ_2^2 distribution and $\widehat{\sigma}$ is a consistent estimator of σ. A natural choice for $\widehat{\sigma}$ is the mean of residual squares of LLK (local linear kernel) estimator of the true intensity function f using the circular neighborhood of size h_n. Moreover, α_n can be specified beforehand to be small number, say, $\alpha_n = 0.01$.

Selection of the bandwidth parameter h_n: in this chapter, we consider choosing h_n using a bootstrap procedure. Larger value of h_n detects several pixels around the JLCs as edge pixels while smaller value of h_n fails to detect several true edge pixels. Therefore, it is a safer option to take a relatively large bandwidth. However, for such cases, the proposed monitoring method will fail to detect a small change near the JLCs. Based on our numerical experience, we suggest choosing $h_n \in [\frac{1.6}{n}, \frac{2.5}{n}]$. In the simulation studies presented in Section 3.7.3, we choose $h_n = \frac{2}{n}$. Readers are referred to Section 3.9.3 for more details about the bootstrap algorithm for selection of h_n.

3.6 Statistical Property

In this section, we discuss some statistical properties of the edge detection procedure mentioned in Section 3.2. The following proposition shows that the estimated jump points converge almost surely to the set of true jump points in *Hausdorff* distance. In our description a point (x,y) is called a *singular* point if one of the following conditions is satisfied:

- There exists some $\rho > 0$ such that, for any $0 < \bar{\rho} < \rho$, the circular neighborhood of

(x, y) with radius $\bar{\rho}$ contains more than two connected regions .

- The jump size is zero.

Other points on JLCs are *non-singular* points. For simplicity, a point on JLCs which has a unique tangent line and non-zero jump is defined as *non-singular* points. Also define,
$\Omega_\epsilon = [\epsilon, 1-\epsilon] \times [\epsilon, 1-\epsilon]$,
$E_\epsilon = \{(x, y) : (x, y) \in \Omega, D((x, y), (x', y')) \leq \epsilon \text{ for some } (x', y') \in E\}$,
$S_\epsilon = \{(x, y) : (x, y) \in \Omega, D((x, y), (x', y')) \leq \epsilon \text{ for some singular } (x', y') \in E\}$,
$\Omega_{\bar{E},\epsilon} = \Omega \backslash E_\epsilon$ and $\Omega_{\bar{S},\epsilon} = \Omega \backslash S_\epsilon$,
where ϵ is a small positive constant, D denotes Euclidean distance, E defines set of points on JLCs. We have defined earlier $\widehat{E}_n = \{(x_i, y_j) : \lambda(x, y) > v_n\}$ as set of detected edge points. Then we have,

Proposition 3.6.1. *Assume that f has continuous first order partial derivatives over $(0,1)\times(0,1)$ except on JLCs at which it has the first order right and left partial derivatives; $h_n = o(1)$, $\frac{1}{nh_n} = o(1)$ and $\frac{log^2(n)}{nh_n^3} = O(1)$; $E|\epsilon|^3 < \infty$; the kernel function is a Lipschitz-1 continuous, isotropic bivariate density function and α_n is so chosen such that $(1-\alpha_n) = o(1)$, $\frac{nh_n^{7/2}}{n^2\sqrt{-log(1-\alpha_n)}} = o(1)$, and $\sqrt{\frac{-log(1-\alpha_n)}{(nh_n)^3}} = o(1)$. Then, for any $\epsilon > 0$, $D_H(E \cap \Omega_{\bar{S},\epsilon}, \hat{E}_n \cap \Omega_{\bar{S},\epsilon}) = O(h_n)$ a.s., where $D_H(A, B)$ is the Hausdorff distance between two point set A and B.*

The above proposition establishes the strong consistency of the detected edge pixels by the aforesaid procedure. Proof of the Proposition 3.6.1 is presented in Appendix A.2.

3.7 Numerical Studies

In this section, we numerically assess the performance of our proposed method in comparison with a number of methods already in the literature. We perform comparative studies on both, artificially created toy images and real images. In the manufacturing industry, any change in intensity values in a region is very common and often considered as faults. However, in many applications such as satellite imaging and medical imaging, these changes should not be taken as meaningful faults or changes in the underlying image process. See Figure 3.3 for this type of situation. In literature, a vast majority of methods are intensity based, and therefore, they are very sensitive to small changes in intensity in a region. Therefore, such methods are very sensitive to noise and as a result, the proportion of false detection by these methods is also large. The proposed method primarily focuses on major image features such as edges and monitors any change of that, rather than just

monitoring any change of intensities. In this regard, the proposed method is particularly useful in monitoring satellite images and medical images.

We discuss the performance for Phase-I and phase-II separately. Here we consider the following methods for numerical comparisons:

- Wavelet-based image monitoring proposed by Koosha et al. (2017)
- Phase-I and Phase-II monitoring of spatial surface data from 3D printing proposed by Zang and Qiu (2018a,b).

The above methods are comparable in the sense that they do not take into account the correlation structure of the images also they assume the independence among the images of different time points.

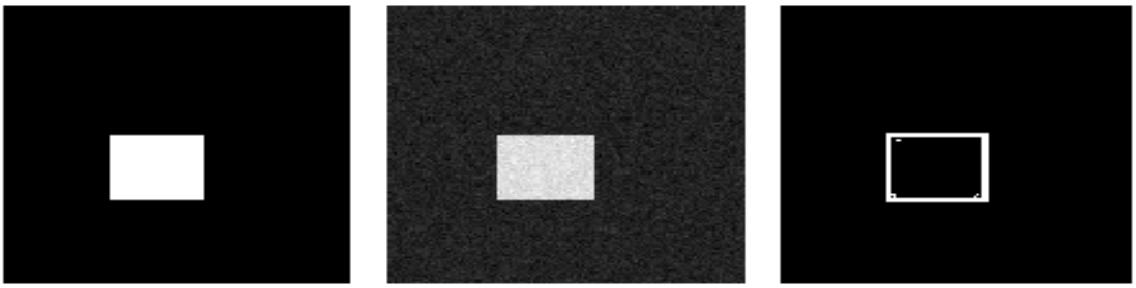


Figure 3.1: (a) True simulated image, (b) Observed Phase-I simulated image, (c) Edge detected image of estimated nominal image.

3.7.1 Brief description of the competing methods:

The wavelet-based method for image monitoring uses a nonparametric profile monitoring approach where each image is decomposed into a set of 1D profiles and monitors them as a whole using a GLR control chart. To extract the frequency domain features, the method applies wavelet transformation and monitors the extracted features over time. In Koosha et al. (2017), wavelet coefficients are termed as frequency domain features and 'Haar' basis function is used to get the wavelets coefficients. Suppose we have m 2-D grayscale images of size $n \times n$. Then each row of an image is considered as 1D profile and suppose there are d estimated features from each profile. Therefore, each image feature is a $nd \times 1$ dimensional vector. In phase-I, there are m images therefore we get m coefficients vector each with dimensions $nd \times 1$. Suppose the coefficient vector of an image is denoted as $\mathbf{\Gamma}^{\intercal} = (\gamma_{11}, ..., \gamma_{1d}, ..., \gamma_{n1}, ..., \gamma_{nd})$. Then, we are interested in detecting the changes of the coefficients over time. To monitor the images in Phase-II, the GLR control chart

statistic by Koosha et al. (2017) for s-th Phase-II image is defined as:

$$R_s = \max_{k\in\Gamma} \frac{s}{\sigma_k^2}\Big(\widehat{\gamma}_{1,s}(k) - \widehat{\gamma}_0(k)\Big)^2, \tag{3.7.1}$$

$s = 1, 2, \ldots$ being the image number in the online monitoring step, $\widehat{\gamma}_{1,s}(k) = s^{-1}\sum_{t=1}^{s}\widehat{\gamma}_{t,k}$, where $\widehat{\gamma}_{t,k}$ is the k-th element of the Γ vector for the t-th sample, $\gamma_0(k)$ is the k-th element of the Γ vector corresponding to nominal image and σ_k^2 is the variance of the k-th coefficient computed from Phase-I images of size m. In simulation studies, nominal images are known to us but in the real life scenario we have to estimate the nominal image from m images of Phase-I images. Koosha et al. (2017) take another parameter and also maximize the the statistic over the parameter to get the estimate of the change-point. However, since we are not interested to estimate the change-point, we modify the statistic like above. The chart statistic will give signal at the s-th image if $R_s > c_R$ where c_R is the threshold value calculated by trial and error method.

Figure 3.2: Different out of control images (without noise).

Now, we discuss another existing method based on Phase-I and Phase-II monitoring of 3D printing [Zang and Qiu (2018a,b)]. This is a potential competing method in the sense that we can always think the image intensity as a 3-D surface, and they also assume spatial independence in the object surface. Here, we assume that images are already geometrically aligned, and if not, then we can use an appropriate image registration technique to register them. Consider the model in the Phase-I step as

$$w_{ijk} = f_k(x_{ik}, y_{jk}) + \varepsilon_{ijk}; \quad \text{for} \quad i, j = 1, 2, \ldots m \quad \text{and} \quad k = 1, 2, \ldots n.$$

where the variables are same as defined above in the equation (3.3.1). Then, the control chart statistic is defined as

$$Q_{ART} = \max_{1\le k\le n} \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} |\widehat{f}_k(x_{ik}, y_{jk}) - \widehat{f}_0(x_i, y_j)|, \tag{3.7.2}$$

where $\widehat{f}_k$ is a jump preserving surface estimate of f_k and $\widehat{f}_0(x_i, y_j) = \frac{1}{n}\sum_{k=1}^{n}\widehat{f}_k(x_{ik}, y_{jk})$

is the estimated intensity value at (i, j)-th position from the images in Phase-I. For more details about the jump preserving surface estimates, readers are referred to Qiu (2009). Zang and Qiu (2018b) use CUSUM statistic in Phase-II stage. However, to make it comparable with our proposed method, we use Shewart-type chart based on the statistic in (3.7.2). The chart will give an out of control signal if $Q_{ART} > c_{ART}$ where c_{ART} is the $(1-\alpha)$-quantile of the empirical distribution of Q_{ART}. If we use L_2 norm instead of L_1 norm in the chart statistic, then it is denoted by Q_{SRT}.

3.7.2 Performance evaluation measures

To evaluate the performance of the proposed method in comparison with the existing methods mentioned above, we use several measures for different phases of monitoring. In the Phase-I stage, to compare the out-of-control (OC) performances, we use two measures: fraction correctly classified (FCC) and false positive proportion (FPP) [Chen et al. (2015)]. The first criterion, named fraction correctly classified (FCC) is defined as the proportion of sample in repeated simulation that are correctly classified. For example, if there are 20 in-control (IC) images and 10 out-of-control (OC) images in a single replication and a specific chart gives out-of-control signal to 5 IC images and 8 OC images, then its FCC is calculated as $\frac{(15+8)}{(20+10)} \approx 0.77$. The second criterion for Phase-I OC performance measure, false positive proportion (FPP), is the proportion of false signals among all signals. In the above example, the FPP is calculated as $\frac{(5+2)}{(5+8)} \approx 0.58$. In Phase-II, performance evaluation is done using 3 different criterion. Average run length (ARL), median run length (MRL) and standard deviation of run length calculates the efficiency of the proposed method and competing method for detecting faults as early as possible. Note that, in the Phase-II stage, we have only the zero-state calculation, i.e., in each simulated process monitoring, OC images start at the first observation time. As the proposed method is a Shewart-type control procedure, in Phase-II the control chart statistic depends only on that image and not on any image in the past. Moreover, we assume that Phase-I samples are independent across different time points. Also, no temporal aggregation is needed as in each time point we only have one single image [Zwetsloot and Woodall (2021)]. Therefore, instead of steady-state calculation, zero-state calculation is a reasonable choice in our context.

3.7.3 Simulations

In this section, we simulate various out-of-control images and compare the performance of our proposed method with other competing methods mentioned above. In our simulation, the true image intensity function of the in-control (IC) image has the expression: $f_0(x, y) = \mathbb{I}(0.31 < x < 0.58, 0.31 < y < 0.54)$, $(x, y) \in [0, 1] \times [0, 1]$, where $\mathbb{I}(.)$ is the

indicator function. The true IC image is shown in Figure 3.1(a). We consider four out-of-control (OC) images and denoted these as OC Image-1, OC Image-2, OC Image-3, and OC Image-4 respectively. To generate Phase-I and Phase-II images, various types of noise such as Poisson, Salt and Peeper or Gaussian noise can be added as a white noise with the nominal image (f_0). Here we added a zero-mean Gaussian distribution with standard deviation (σ) equaling 0.05. For determining the correct level of noise, signal-to-noise ratio (SNR) is an important measure (c.f., Koosha et al. (2017)). Figure 3.2 shows the simulated OC images. The size of all of these simulated images is 128×128 (i.e, n=128). When noise level is high, our edge detection method falsely detects several edge pixels in the background of the images. To get rid of those falsely detected edge pixels, we run the following adjustment procedure:

Algorithm for removing false edges

1: $\mathbf{E}^{n\times n}$ = **Binary Edge Matrix**
2: $\mathbf{E_p}^{(n+2p)\times(n+2p)}$ = **Padded Edge Matrix (Pad Length= p)**
3: **for** $i := (1+p)$ to $(n+p-1)$ **do**
4: **for** $j := (1+p)$ to $(n+p-1)$ **do**
5: **if** $\mathbf{E_p}[i,j] \leftarrow 1$ & $\sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1}\mathbf{E_p}[k,l] \leq 4$ **then**
6: $\mathbf{E_p}[i,j] \leftarrow 0$
7: **end if**
8: **end for**
9: **end for**

Note that for elements near the boundaries of the image matrix we use the mirroring technique for which there is no subscript out of the bound situation for the above algorithm. Table 3.1 shows the control limits for the methods to achieve in-control ARL, i.e., $ARL_0 \approx 50$ when the size of Phase-I data are $m = 10$ and $m = 20$. Note that the values of m considered in this study is relatively small compared to the conventional SPC literature, because in most of the applications of medical diagnostics, satellite imaging, etc., the number of images of the same object is always a few, in the order of 10's or even smaller in some cases. Therefore, it is reasonable to construct the control chart based on small number of Phase-I samples and correspondingly small pre-fixed ARL_0. To find the empirical distribution, bootstrap technique could be used. However, to set the control limit at the prefixed ARL_0 value we use trial and error methods. Readers are referred to Section 3.4 for more details to determine the control limit with a pre-fixed ARL_0 value. Here, D_{new} corresponds to the proposed method , R_k is the statistic (3.7.1) based on the wavelet-based technique, and Q_{ART} corresponds to the surface monitoring method defined

in (3.7.2). From the definition of FCC and FPP, we expect a large value of FCC and a small value FPP if the monitoring method works well.

n	D_{New}	Q_{ART}	Q_{SRT}	R_k
10	1.414214	0.042375	0.0028167	115.50
20	1.414214	0.041349	0.0026817	33.000

Table 3.1: Control limits of the charts when m=10 or 20 and $ARL_0 \approx 50$.

FCC and FCC measure the performance of the methods in Phase-I step. In this study, to calculate the FCC and FPP values, we consider 500 simulated in-control images and 500 simulated out-of-control images. From Table 3.2, we observe that for each type of image, D_{new} has large FCC and small FPP values. OC Image-3 deviates only a little from the in-control image, and still, our proposed method performs well. There is a substantial difference in FCC and FPP values with other competing methods that ensures the effectiveness of our proposed method. Moreover, for OC Image-3, neither of the competing methods work well in Phase-II stage, but our proposed method still gives an excellent result. The advantage of the proposed method is that if there is a prominent change even in a very small region of an image, it is capable to detect.

Image type	Method	FCC	FPP	ARL_1	$me(ARL_1)$	$sd(ARL_1)$
OC-Image-1	D_{new}	0.9906667	0.02723735	1.00	1.00	0.00
	R_k	0.9886667	0.03288201	1.00	1.00	0.00
	Q_{ART}	0.9893333	0.03100775	1.00	1.00	0.00
OC Image-2	D_{new}	0.9906667	0.02723735	1.00	1.00	0.00
	R_k	0.9886667	0.03288201	1.00	1.00	0.00
	Q_{ART}	0.9893333	0.03100775	1.00	1.00	0.00
Oc Image-3	D_{new}	0.9906667	0.02723735	1.000	1.000	0.0000000
	R_k	0.8186667	0.06367041	5.920	6.000	0.6127621
	Q_{ART}	0.6753333	0.3695652	19.26	14.000	18.434250
Oc Image-4	D_{new}	0.9906667	0.02723735	1.00	1.00	0.00
	R_k	0.9886667	0.03288201	1.00	1.00	0.00
	Q_{ART}	0.9893333	0.03100775	1.00	1.00	0.00

Table 3.2: Phase-I and Phase-II performances for OC Images

Now, we are going to discuss various scenarios that ensure the effectiveness of our proposed method over the competing methods. It shows why edge based image surveillance is important. The IC image is shown in Figure 3.1.

Case A: Here, the image object is same but the contrast is changed, see Figure 3.3(a) for this scenario. This type of situation is very common in real life. For instance, we capture an image of an object with no shadow on it, and then capture another image from the same position in the presence of shadow in the background or on the image ob-

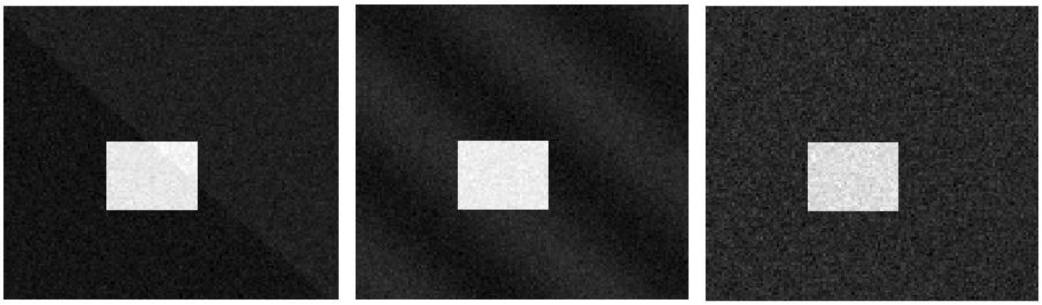


Figure 3.3: Different in-control images. (a) Case A is a situation where IC image has been taken in the presence of light shadow, (b) Case B represent similar situation where image object remains same but continuous change in background, and (c) Case C presents the same image with slightly higher noise level.

ject itself. Both images will be the same but there would be changes in the intensity values.

Case B: In this situation, the image object remains the same but there is a smooth change in the background. For instance, we capture an image in the presence of diffused light in the background. Figure 3.3(b) presents one such example.

Case C: Here also, the noise level of the images are different from the noise level of the IC image. It is very relevant in the real life scenario because it is not always possible to get the images at a fixed level of noise. In our example, the IC image in Figure 3.1(b) has noise level $\sigma = 0.05$ and the image in Figure 3.3(c) has noise level 0.06.

Image type	Method	FCC	ARL_1	$me(ARL_1)$	$sd(ARL_1)$
Case A	D_{new}	0.98133	30.330	20.00	29.07694
	R_k	0.65800	1.135	1.00	0.3425811
	Q_{ART}	0.65533	1.00	1.00	0.00
Case B	D_{new}	0.98867	43.825	38.000	35.48012
	R_k	0.49600	1.0000	1.0000	0.00
	Q_{ART}	0.47600	1.0000	1.0000	0.00
Case C	D_{new}	0.98533	41.825	33.000	35.330
	R_k	0.98600	46.270	48.000	15.48243
	Q_{ART}	0.65533	1.0000	1.0000	0.00

Table 3.3: Phase-I and Phase-II performances for cases A,B and C.

Since the image object is the same for both cases, and therefore, the images are in-control, and ARL_1 should be the same as ARL_0. From Table 3.3, it is clear that in all the cases mentioned above, proposed method performs well based on the large FCC value and ARL_1 value. The wavelet-based method performed well for the last case but the surface-based method is not appropriate for these types of situations. Therefore, despite

good performances in certain situations, we should be more careful to use the competing methods in practical applications.

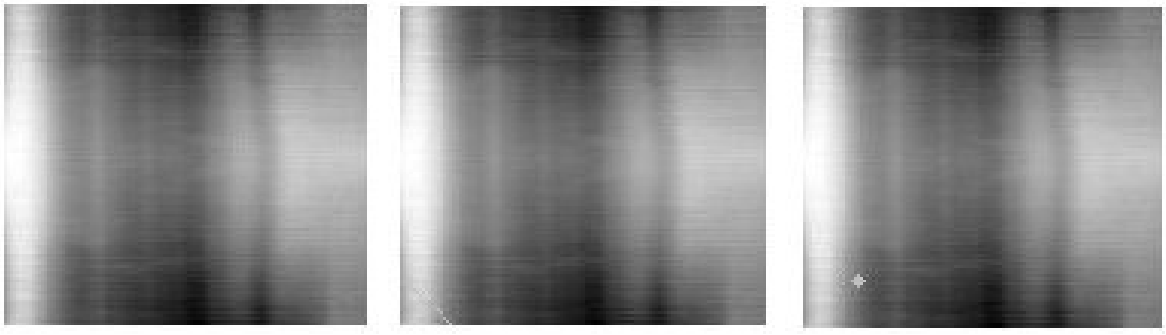


Figure 3.4: (a) Smooth steel surface, (b) Steel surface with scratch, (c) Steel surface with circular fault.

Next, we perform a comparison study on real images from a production process. We consider the images of smooth steel surface as IC images and insert various types of simulated faults. Referred to Figure 3.4(a) for an IC image. This is a grayscale image and we resize it to 128×128. To artificially create Phase-I data, we add Gaussian noise with mean zero and standard deviation 0.001. Table 3.4 shows the control limits for different Phase-I sample sizes.

n	D_{New}	Q_{ART}	Q_{SRT}	R_k
10	11.18034	0.00084534	0.0000011215	58.00
20	11.18034	0.00082480	0.0000010677	20.00

Table 3.4: Control limits of charts when m= 10 and 20 for $\alpha = 0.1$.

For the comparison study in Phase-II, we incorporate various types of shifts to create OC images. In Figures 3.4(b) and 3.4(c), we consider scratch and circular faults, respectively on the steel surface which are most common in the manufacturing industry. Table 3.5 shows the performance measures for this example where in-control ARL is 20. For the steel surface image with a scratch, all methods perform well based on the out-of-control ARL measures, but based on FCC and FPP, the surface-based method and the proposed method performs relatively better than the wavelet-based method. However, for the image with the circular-fault, the proposed method is not up to the mark as compared to the competing methods. This is mainly due to false edge detection. A major limitation of this method is that it is highly affected by false jumps in the background. This is because of the fact that the *Hausdorff* distance is very sensitive to individual points in the related point sets.

Image type	Method	FCC	FPP	ARL_1	$me(ARL_1)$	$sd(ARL_1)$
Scratch	D_{new}	0.9673333	0.08925319	1.00	1.00	0.00
	R_k	0.9180000	0.19743180	1.00	1.00	0.00
	Q_{ART}	0.9693333	0.08424908	1.00	1.00	0.00
Circular-Fault	D_{new}	0.8033333	0.16171620	1.985	1.00	1.5184
	R_k	0.9180000	0.19743180	1.000	1.00	0.0000
	Q_{ART}	0.9693333	0.08424908	1.000	1.00	1.0000

Table 3.5: Phase-I and Phase-II performance for Steel Surface

3.8 Real Data Example

In this section, we consider a real image example where we are interested in monitoring brain tumor growth [Stamatakos and Giatili (2017)]. The panels of Figure 3.5 show tumor growth at 0-th day, 60-th day, 120-th day, and 180-th day, from left to right. As a pre-processing step, we convert the original color images to grayscale images. Moreover, we resize each image to the resolution of 128×128 and rescale the image intensities so that their values are in the range 0 to 1.

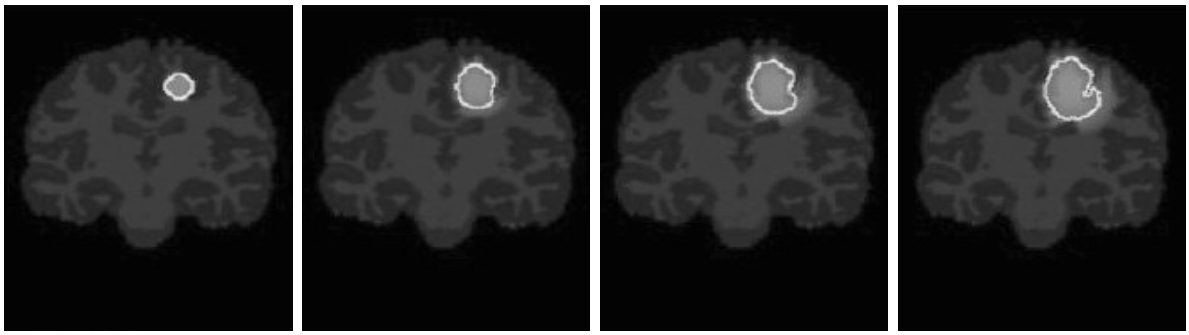


Figure 3.5: Visualization of a virtual glioblastoma tumor growth in vivo on a coronal slice at various time points

In this study, we consider the 0-th day brain image as an IC image and now we have three different cases. For the first case, our objective is to check the ability to detect the change if we have 180-th day image as an OC image. Similarly, the other two cases are either we consider 120-th day or 60-th day image as OC image. For each case, we have calculated the FPP and FCC value and corresponding out-of-control ARL value, denoted as $ARL_{OC}^{(k)}; k = 180, 120, 60$. To generate Phase-I and Phase-II images, we add a zero mean Gaussian noise with standard deviation of 0.004. Table 3.6 presents the control limits for the competing methods when the number of in-control Phase-I images are 10 and 20, respectively. In this example, we fix the in-control ARL value at 10 and to calculate the control limits, we replicate each set of the simulations 200 times.

From Table 3.6, it is clear that for a change in the value of m for our proposed method change of control limit is negligible whereas for the other methods significant amounts of

m	D_{New}	Q_{ART}	Q_{SRT}	R_k
10	5.2	0.003374	0.00001786	35.00
20	5.1	0.003292	0.00001700	29.50

Table 3.6: Control limits of charts when m = 10 and 20, for $\alpha = 0.1$.

changes are evident. Now, in Table 3.7 we have shown different metrics for evaluating performance. Since the noise level is not high, we have not done any algorithm for false edge removal in the background. Because there is no substantial change in the control limit for all the calculations we use samples of size m=10. Since the amount of change in images over time is respectively large, all the methods perform very well and give perfect detection. But, if we look into the FCC and FPP value, the proposed method gives a high FCC and low FPP value which is very important in medical science. The wavelet-based method performs comparatively well over the surface-based method. To calculate the FCC and FPP, we have taken 1000 in-control images and 1000 out-of-control images. In the Phase-II stage, for each set of Phase-II sample we calculate the out-of-control ARL and to calculate the mean, median and standard deviation of the out-of-control ARL, we replicate the process for 100 set of Phase-II samples. Since the changes in the image is relatively large, all methods perform well. However, based on Phase-I metrics, the proposed method outperforms its competitors.

	k-th Image	FCC	FPP	$ARL_{OC}^{(k)}$	$me(ARL_{OC}^{(k)})$	$sd(ARL_{OC}^{(k)})$
D_{new}	k=60	0.9386667	0.1554054	1.00	1.00	0.00
	k=120	0.9386667	0.1554054	1.00	1.00	0.00
	k=180	0.9386667	0.1554054	1.00	1.00	0.00
Q_{ART}	k=60	0.7673333	0.4110718	1.00	1.00	0.00
	k=120	0.7673333	0.4110718	1.00	1.00	0.00
	k=180	0.7673333	0.4110718	1.00	1.00	0.00
R_k	k=60	0.9380000	0.1568297	1.00	1.00	0.00
	k=120	0.9380000	0.1568297	1.00	1.00	0.00
	k=180	0.9380000	0.1568297	1.00	1.00	0.00

Table 3.7: Phase-I and Phase-II performance for the competing methods.

3.9 Guide for Practitioners

3.9.1 Change point estimation

Post-signal diagnosis is an important task in any process monitoring procedure. A major step in post-signal diagnosis is to accurately estimate the time point (τ) at which the process becomes OC, i.e., to estimate change point of the process carefully. The proposed

image monitoring method being a Shewhart-type procedure, $\widehat{\tau}$, an estimate of the change point, is given by the first time point at which the chart indicates a signal. The proposed method is capable of estimating the change point almost accurately as indicated in Table 3.2. Table 3.8 presents the performance of the change point estimation. In Phase II monitoring, we set the change point by adding a sustained shift at time point $\tau = 6$ and onward. That is, an OC image is incorporated in the process at time point 6 and onward. To calculate the mean and standard deviation of the estimated change point we replicate the process for 100 sets of Phase-II sample sequences. We have shown the change point estimation for OC Image-1 and OC Image-3 and the results are consistent with the out-of-control ARL value in Table 3.2. From the table, it is clear that we have a perfect result. This is because of the fact that we remove the false edges in the background due to noise in the image using the algorithm in Section 3.7.3. Therefore, as long as the noise level of the image is moderate and there is a prominent change in the image object, the proposed method is capable of detecting the change and estimating the change point almost accurately.

Image	$mean(\widehat{\tau})$	$sd(\widehat{\tau})$
OC Image-1	6.00	0.00
OC Image-3	6.00	0.00

Table 3.8: Performance of change point estimation.

3.9.2 Detection of fault regions

In the case of monitoring images, after an OC signal is detected, our main objective is to diagnose the exact fault regions in those OC images so that corrective measures are taken quickly. The proposed method for monitoring images is equivalent to monitoring the corresponding JLCs of the images. For the ℓ-th online image, if there is a signal then we define the possible fault regions as $(\widehat{E}_\ell \setminus \widehat{E}_0) \cup (\widehat{E}_0 \setminus \widehat{E}_\ell)$, where $A \setminus B$ is defined as the set difference between two sets A and B. In case of high-resolution images, Proposition 3.6.1 indicates theoretical justification for finding possible fault regions. When the noise level is high in an image, there are often many falsely detected fault regions. To remove those, we use the same algorithm as in Section 3.7.3. Figures 3.6(a-d) present the performances of the proposed fault region estimation. Figures 3.6(a) and 3.6(b) show the actual change and estimated change locations for OC Image-1, discussed in Section 3.7.3. Similarly, Figures 3.6(c) and 3.6(d) are for OC Image-3. Figures 3.6(a-d) indicate highly accurate detection of the fault regions by the proposed technique. A major advantage of the proposed fault region detection technique is that it can detect multiple fault regions in an OC image. See Figure 3.6(a) for multiple fault regions.

Figure 3.6: (a) Actual change for OC Image-1, (b) Estimated change regions for OC Image-1, (c) Actual change for OC Image-3, and (d) Estimated change region for OC Image-3.

3.9.3 Bootstrap algorithm for bandwidth selection

Selection of h_n is an important task in the proposed edge-based monitoring procedure. Here we propose the detailed algorithm based on bootstrap procedure. In order to choose the bandwidth parameter, we use Hausdorff metric again to calculate the distance between the point set of the true jump points and the estimated jump points. Note that, a bandwidth parameter is preferable for which this distance is small. Figure 3.7, shows a plot of the distance between the estimated point set and the true jump points for different values of h_n, with the reference image in Figure 3.1. Based on the above diagram we suggest choosing $h_n \in [\frac{1.6}{n}, \frac{2.5}{n}]$. In particular, $h_n = \frac{2}{n}$ works well in most applications. Readers are referred to Mukherjee and Qiu (2011) for selecting h_n by a cross-validation method.

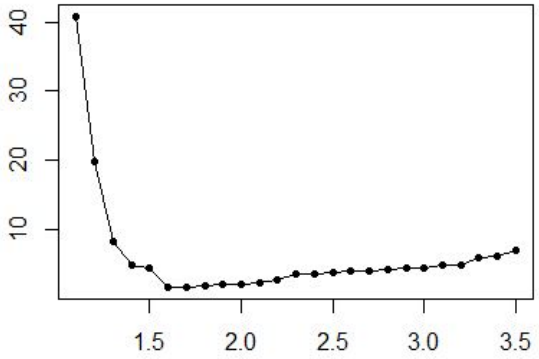


Figure 3.7: *Hausdorff* distance between point set of estimated jump points and true jump points for different values of h_n.

Algorithm:

Step-1: Apply local piece-wise smoothing procedure for image surface estimation and calculate the residuals $\{\widehat{\varepsilon}^*\}$ and $\widehat{\sigma}$ from the reference image.

Step-2: Generate a new image by adding a Gaussian noise with $N(0, \widehat{\sigma}^2)$.

Step-3: To find the estimated jump points, use the proposed jump detection algorithm in Section 3.2, for a fixed level of h_n.

Step-4: Repeat Steps 2 and 3 B times, where B is a sufficiently large number. The sets of detected jump points from the bootstrap samples are denoted as $\widehat{E}^*_{1n}, \widehat{E}^*_{2n}, \ldots, \widehat{E}^*_{Bn}$, respectively. Then, the bootstrap estimator of the distance between the estimated and the true jump points $(D_H^{BT}(\widehat{E}_n, E; h_n))$ is defined as

$$\widehat{D_H^{BT}}(\widehat{E}_n, E; h_n) = \frac{1}{B}\sum_{k=1}^{B} D_H(\widehat{E}^*_{kn}, E; h_n)$$

The optimal bandwidth is then approximated by minimizing $\widehat{D_H^{BT}}(\widehat{E}_n, E; h_n)$, with respect to h_n.

3.10 Conclusion

In this chapter, we propose a new procedure for monitoring image data. This is a Shewhart-type control chart for monitoring image data based on detected edges. The method is very simple to construct and convenient to use even in the high-dimensional problem of image monitoring. Similar to the Shewhart control chart, the proposed control chart monitors the process based on the observed images at the one-time point. Therefore, it is useful to detect relatively large and transient shifts in the images but not designed for detecting a small persistent shift. This method is capable enough to ignore smooth insignificant changes in the image surface. However, a method that can capable of ignoring any type of insignificant change is desirable. A more flexible image comparison and monitoring scheme based on JLCs has been discussed in the next chapter.

Chapter 4

Image Comparison Based on Local Pixel Clustering[†]

4.1 Introduction

Purpose of this chapter is to propose an image comparison algorithm that is more robust in presence of insignificant changes in the image intensity function. Here we propose a novel and effective image comparison procedure based on edge information and other fine details of an image object. One major advantage is that it does not *directly* depend on the intensity values of the image and also, does not require an explicit edge detection algorithm, and hence the method is not very sensitive to background noise of the images. Moreover, the complicated edge structures can be compared efficiently by a local pixel clustering algorithm. The steps for our image comparison methods are as follows. Firstly, for each image we cluster the pixels in a neighborhood into two groups based on their intensity values, and thereafter compare the clustering outcomes from each image by a cluster based metric. In the literature, there are several similarity and dissimilarity measures for comparing clustering outcomes. Some of those include, measure based on counting pairs, measure based on mutual information, etc. One major drawback of these measures is that they do not take into account the spatial information. As the image surface has spatial nature, therefore, along with the proportion of types in a group of the clustering, their positions are equally important to address. In the proposed methodology, we choose *Variation of Information* (VI) [Meilă (2007)] as a metric between two clustering outcomes, and construct a test based on this metric, and thus all issues mentioned above can be addressed properly. Moreover, the proposed method is robust to the presence

[†]This chapter is based on the publication Roy and Mukherjee (2024b): **Roy, A.** and Mukherjee, P.S. "Image Comparison Based On Local Pixel Clustering", *Technometrics*, 2024, 66(4), 495–506. DOI: 10.1080/00401706.2024.2322670.

of minor insignificant changes in the background. Also, adjusting the value of a tuning parameter, the proposed method is capable of detecting small to large changes, based on the purpose of the applications. For demonstration, let us examine the toy image in Figure 4.1. The image object in the left and right panels are the same; however, the intensity functions in the background and also in the foreground are different. This situation is common in many real-life applications where there are possibilities of shadows on the image object. As the vast majority of methods in the existing image monitoring literature rely only on changes in image intensities to identify out-of-control images, they will consider it a change. However, the proposed algorithm is capable to ignore this minor alternation of image intensity function in the background and foreground. For a similar discussion on satellite imaging, readers are referred to Section 4.6. Hence, it is more flexible and useful in real life scenarios. Both theoretical arguments and numerical studies show that the proposed method is indeed an effective tool for image comparison and monitoring.

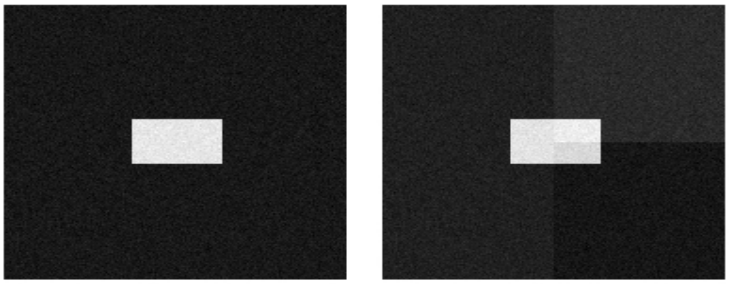


Figure 4.1: A demonstration on the advantage of the proposed algorithm.

4.2 Proposed Methodology

As discussed earlier, under the JRA literature [Qiu (2005)], a 2-D grayscale image can be considered as a discontinuous regression surface, where edges of the image object are considered as discontinuity points or jump points. The locations of the discontinuity points are considered as JLCs. Essentially, we aim to frame an image comparison procedure based on these JLCs of the images. Although the proposed method is useful for various kinds of images, we only consider 2-D monochrome images in this article.

4.2.1 Model description

In the JRA literature, any $n \times n$ image can be expressed as the following 2D JRA model [Mukherjee and Qiu (2015)]:

$$w_{ij} = f(x_i, y_j) + \varepsilon_{ij} \ , \quad \text{for} \quad i, j = 1, 2, \ldots, n, \tag{4.2.1}$$

where $\{(x_i, y_j) : i, j = 1, 2, ..., n\}$ are equally spaced design points (or pixels) in the design space $\Omega = [0, 1] \times [0, 1]$, $f(x, y)$ is an unknown image intensity function at (x, y), $N = n^2$ is the sample size, and ε_{ij} are independent and identically distributed (i.i.d.) random errors with mean 0 and variance $\sigma^2 > 0$. Note that, the unknown image intensity function f is assumed to be continuous except on the boundaries of the image objects. In the JRA literature, these boundary curves are popularly known as jump location curves (JLCs). The readers are referred to the A.3 for mathematical description of the JLCs.

4.2.2 Image registration

Geometric misalignment of images is common in image monitoring applications. Due to the change in the relative position of camera and image objects at different time points, sometimes, the images of the same production line could be mismatched. Hence, to monitor the images automatically by the computer, they should be matched up first. In image processing literature, this is commonly known as *image registration* [Avants et al. (2008)]. The main objective of any image registration procedure is to estimate the geometric transformation to geometrically match one image to another. In the literature, both parametric [Zhang et al. (2020)] and nonparametric [Xing and Qiu (2011)] transformations have been considered. One can use any existing method available in the literature for performing image registration as a pre-processing step before applying our proposed image comparison method. For a reliable image comparison, it is crucial that the images are registered properly. Geometric misalignment of images affects the performance of the proposed algorithm and a good registration procedure makes our comparison method more accurate even for a small change.

4.2.3 Local pixel clustering

In this chapter, our main concern is to compare two images based on image features. JLCs are important features of an image. In this step, we consider local pixel clustering to get information about JLCs in each circular neighbourhood. Since the observed image intensity values are scalars, the first step is a 1-D clustering problem where the maximum

number of clusters is two. In real life images, including satellite and medical images, the case where more than two image segments intersect in a small circular neighbourhood of a pixel is very rare. Therefore, clustering the pixels in a circular neighborhood into two groups based on the intensity values is a natural choice.

For a given pixel $(x, y) \in \Omega$, let us consider the circular neighborhood

$$B(x, y; h_n) = \{(u, v) : (u, v) \in \Omega, \sqrt{(u-x)^2 + (v-y)^2} \leq h_n\},$$

where $h_n > 0$ is a bandwidth parameter. Now, our objective is to cluster the pixels into groups based on the observed intensity values w_{ij}'s in $B(x, y; h_n)$. Theoretically, any reasonable clustering method can be used. In this chapter, we have a simple 1-D classification problem, and we use a simple but effective pixel clustering suggested by Mukherjee and Qiu (2015). Such a pixel clustering procedure is reasonable in our context due to the fact that it can reflect local edge structures and preserve them well, while smoothing without imposing restrictive conditions on the smoothness or shape of the JLCs. The central idea is based on within-group and between-group variability of the intensity values within a circular neighborhood. If $B(x, y; h_n)$ indeed contains two continuity regions of the image intensity function, then the ratio of between-group variability and within-group variability would be large. On the other hand, if $B(x, y; h_n)$ contains one continuity region, it implies a small value of the ratio. Therefore, we use this ratio as a criterion based on which we decide whether to classify the local intensity values into two groups or not. To classify the pixels in $B(x, y : h_n)$ into two clusters, we introduce c as the cut-off constant and define two groups as

$$B_1(x, y; h_n, c) = \{(x_i, y_j) : (x_i, y_j) \in B(x, y; h_n) \text{ and } w_{ij} \leq c\}$$

$$B_2(x, y; h_n, c) = \{(x_i, y_j) : (x_i, y_j) \in B(x, y; h_n) \text{ and } w_{ij} > c\},$$

where $c \in \mathcal{I}(x, y; h_n) = \left(\min_{(x_i,y_j)\in B(x,y;h_n)} w_{ij}, \max_{(x_i,y_j)\in B(x,y;h_n)} w_{ij} \right)$. Therefore, both $B_1(x, y; h_n, c)$ and $B_2(x, y; h_n, c)$ are nonempty sets. Moreover, C_0, the optimal value of c can be approximated by the following optimization problem:

$$\max_{c\in\mathcal{I}(x,y;h_n)} \frac{|B_1(x, y; h_n, c)|(\bar{w} - \bar{w}_1)^2 + |B_2(x, y; h_n, c)|(\bar{w} - \bar{w}_2)^2}{\sum_{(x_i,y_j)\in B_1(x,y;h_n,c)} (w_{ij} - \bar{w}_1)^2 + \sum_{(x_i,y_j)\in B_2(x,y;h_n,c)} (w_{ij} - \bar{w}_2)^2}, \tag{4.2.2}$$

where $\bar{w}$, $\bar{w}_1$ and $\bar{w}_2$ are the sample averages of the image intensities in $B(x, y; h_n)$, $B_1(x, y; h_n, c)$ and $B_2(x, y; h_n, c)$, respectively, and $|A|$ denotes the number of elements in A. Since the number of pixels in a circular neighborhood $B(x, y; h_n)$ is finite, the optimal choice C_0 always exists. To this end, the above algorithm (4.2.2) for selection of the

optimal threshold value is very similar to Otsu's algorithm for image thresholding [Otsu (1979)]. Note that, if $B(x, y; h_n)$ is within a continuity region, then local clustering of that neighborhood is not important. Therefore, to increase the efficiency of the proposed method, it is necessary to check whether clustering is important in that neighborhood. Define $S(x, y, h_n)$ to be the sample standard deviation of all w_{ij}'s in $B(x, y; h_n)$. Intuitively, if $S(x, y, h_n)$ becomes larger than σ, it implies the presence of edge curves in the neighborhood. However, in our proposed method, we perform local clustering if

$$S(x, y, h_n) > \kappa\sigma, \tag{4.2.3}$$

where, κ is a procedure parameter. In practice, σ is not known and has to be estimated by some suitable procedure. One simple estimator of σ is

$$\widehat{\sigma} = \sqrt{\frac{1}{n^2}\sum_{i,j}(w_{ij} - \widehat{f}(x_i, y_j))^2}, \tag{4.2.4}$$

where $\widehat{f}$ is a jump preserving estimator of the image intensity function. In literature, there are extensive discussions about the jump preserving surface estimation procedure. Readers are referred to [Qiu (2009)] for more discussions about jump preserving surface estimates.

4.2.4 Image comparison

One fundamental step behind any image monitoring procedure is image comparison. In this chapter, we propose an image comparison method based on local clusters. Under the assumption that two images of the same object are geometrically aligned, any image comparison procedure involves testing the hypotheses

$$H_0 : \text{True image intensity functions of the two images are same}$$
$$\text{vs.}$$
$$H_1 : \text{True image intensity functions of the two images are different}$$

However, in certain types of applications, minor changes in the image intensity values should not be considered as a change when the locations of the edges remain unchanged. Therefore, it is more appropriate to test the equality of JLCs rather than equality of the two intensity functions. As the information about the JLCs are captured very well by local pixel clustering, we use these clusters to test the equality of the JLCs.

Variation of Information for comparing local clustering outcomes

In this chapter, we choose *Variation of Information* (VI) as a measure of distance between two clustering outcomes, and construct a test statistic based on that. In the literature, existing popular methods for comparing clustering outcomes are based on set matching [Meilă and Heckerman (2001)], counting pairs of points that agreed or disagreed in the clusters [Fowlkes and Mallows (1983); Rand (1971)], mutual information, etc. However, major drawbacks of these measures are: they can not take into account the spatial structures in the clustering and they are not a true metric. For clarification, let us look at the example in Figure 4.2. The image presents three clustering outcomes with two groups, one group denoted by big circles and another group denoted by small circles. It is clear from Figure 4.2 that the proportion of elements in each group for the left panel and the right panel are same but the positions of the circles have changed. For the middle panel, the proportions have changed. Most of the existing methods are capable of detecting the change between the left and middle, also between the right and the middle. As an image intensity function is spatial in nature, change in the position of elements in the clustering should also be considered as a change in the image intensity function. Since VI can detect such changes in the positions, it is a natural choice for comparing clustering outcomes. Moreover, VI has another major advantage over the others because it enjoys all properties of a metric.

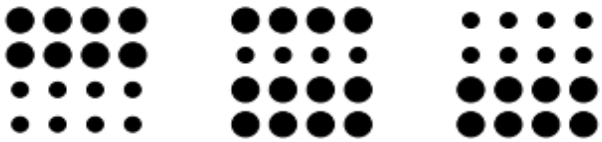


Figure 4.2: Three different clustering outcomes.

The formal definition of VI is as follows. Suppose, $\mathcal{C}$ and $\mathcal{C}'$ are the partitions (clustering) of a set of points or a data-set D. Then, VI for two clustering outcomes can be defined as

$$VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}'), \tag{4.2.5}$$

where, $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropy associated with clustering $\mathcal{C}$ and $\mathcal{C}'$, respectively, and $I(\mathcal{C}, \mathcal{C}')$ is the *mutual information* between two clustering outcomes. Figure 4.3 describes the definition of the VI metric in (4.2.7). The magnitude of the VI metric is determined by the sum of the shaded regions of Figure 4.3. Using entropy, we aim to capture the uncertainty associated with each clustering. Mutual information is useful to asses mutual dependence between two random variables. In this context, mutual information between two clustering outcomes quantifies the information that one clustering

has about another. Readers are referred to Meilă (2007) for more details about the VI metric.

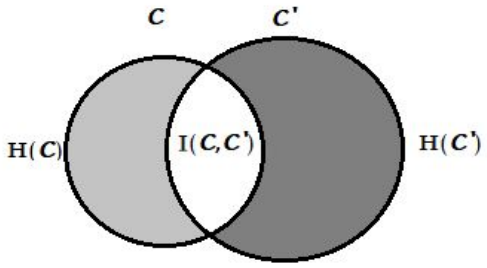


Figure 4.3: Graphical demonstration of the VI metric.

Calculation of VI: Let $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ be a partition of $\mathcal{D}$ such that $C_\ell \cap C_m = \phi$ and $\cup_{\ell=1}^{M} C_\ell = \mathcal{D}$ i.e., mutually exclusive and exhaustive partition of the set $\mathcal{D}$. Let the number of elements in the ℓ-th cluster be q_ℓ, and hence $\sum_{\ell=1}^{M} q_\ell = |\mathcal{D}| = q$ (say). Second partition $\mathcal{C}'$ of the set $\mathcal{D}$ can be expressed similarly with M' clusters where the size of the ℓ'-th cluster being q'_ℓ. Then the entropy associated with $\mathcal{C}$ can be expressed as

$$H(\mathcal{C}) = -\sum_{\ell=1}^{M} P(\ell) log P(\ell),$$

where $P(\ell) = q_\ell/q$. $H(\mathcal{C}')$ can be calculated in a similar way. Moreover, the mutual information can be calculated as

$$I(\mathcal{C}, \mathcal{C}') = \sum_{\ell=1}^{M} \sum_{\ell'=1}^{M'} P(\ell, \ell') log \frac{P(\ell, \ell')}{P(\ell)P(\ell')},$$

where $P(\ell, \ell') = \frac{|C_\ell \cap C'_{\ell'}|}{q}$. Finally, one can calculate the VI metric using (4.2.5).

Proposed test based on VI

Under the 2-D JRA model, k-th image can be expressed by the following regression set-up:

$$w_{ij}^{(k)} = f_k(x_i, y_j) + \varepsilon_{ij}^{(k)}\ ,\ k = 1, 2,\ \ i, j = 1, 2, \ldots, n, \tag{4.2.6}$$

where (x_i, y_j) is the (i, j)-th pixel coordinate, f_k is the true image intensity function of the k-th image, $N = n^2$ is the sample size, $\{\varepsilon_{ij}^{(1)}\}$ and $\{\varepsilon_{ij}^{(2)}\}$ are independent and identically distributed (i.i.d.) random errors with mean 0 and variance σ_1^2 and σ_2^2, respectively. In (4.2.6), we further assume that $\sigma_1^2 = \sigma_2^2 = \sigma^2$, and the images are registered properly.

Here we propose the test statistic:

$$T_{VI} = \frac{1}{n^2} \sum_{i,j} D_{VI}(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j)), \tag{4.2.7}$$

where $\mathcal{C}(x_i, y_j)$ and $\mathcal{C}'(x_i, y_j)$ are the clustering outcomes of the observed image intensities in the circular neighborhood around the point (x_i, y_j) for these two images, respectively, and D_{VI} is the VI-based distance between the two clustering outcomes. It is very challenging to derive the asymptotic distribution of the proposed test statistic. Note that the statistic is not exactly pivotal. Intuitively, the value of the test statistic depends on the misclassification associated with the local clustering outcomes due to the presence of noise in the images. As the sizes of the local clusters depend on the true image intensity function, the statistic is not pivotal. Under H_0, the value of T_{VI} would be small. Otherwise, its value would be large. Moreover, the true null distribution of the test statistic T_{VI} is unknown. See Section 4.2.4 for determining the critical value of the test statistic based on the empirical distribution of the statistic under the null hypothesis. Note that, the statistic T_{VI} is a bounded random variable, and in our context, as the maximum number of clusters in each neighborhood is two, as per the results from Meilă (2007), $T_{VI} \leq 2 \log 2$.

Construction of the empirical distribution of the test statistic

In practice, it is a challenging task to find the asymptotic distribution of T_{VI}. We use a resampling method, popularly known as wild bootstrap technique to calculate the critical value of the proposed test statistic. As recommended by Mammen (1993), wild bootstrap technique has been used to sample from the residuals of the regression model. For similar discussions regarding wild bootstrap, the readers are referred to Zhao et al. (2020); Wang and Ye (2010). Under the null hypothesis, $f_1(x, y) = f_2(x, y) = f(x, y)$, i.e., the true image intensity functions are identical, and the jump preserving surface estimate in the combined sample could be obtained by $\widehat{f}(x, y) = \frac{1}{2}\left(\widehat{f}_1(x, y) + \widehat{f}_2(x, y)\right)$, where $\widehat{f}_1(x, y)$, and $\widehat{f}_2(x, y)$ are the estimated image intensity functions of the two images. In our studies, we use the jump preserving surface estimate proposed by Kang et al. (2018). The null distribution of the test statistic is approximated by the bootstrap distribution of the test statistic. The description of the algorithm follows.

1. Using the combined estimate $\widehat{f}$, construct the residuals $\{\widehat{\varepsilon}(x_i, y_j) = w_{ij} - \widehat{f}(x_i, y_j) : i, j = 1, 2, 3, \ldots, n\}$.

2. Generate two bootstrap images by

$$w^{1*}(x_i, y_j) = \widehat{f}(x_i, y_j) + \widehat{\varepsilon}^*(x_i, y_j),$$
$$w^{2*}(x_i, y_j) = \widehat{f}(x_i, y_j) + \widehat{\varepsilon}^*(x_i, y_j),$$

where $\widehat{\varepsilon}^*(x_i, y_j)$ is a bootstrap sample obtained from the set $\{\widehat{\varepsilon}(x_i, y_j) : i, j = 1, 2, \ldots, n\}$.

3. Calculate the value of VI based test statistic based on the bootstrapped images.

4. Repeat Steps 2 and 3 B times, and use the generated values to find the empirical distribution of the test statistic.

By the above bootstrap algorithm, we can obtain the empirical distribution of the proposed test statistic, and for a level α test, the null hypothesis is rejected if T_{VI} is greater than the corresponding $(1-\alpha)$-th sample quantile of the bootstrap distribution of T_{VI}.

4.3 Practical Guidelines On Parameter Selection

In the proposed image comparison procedure, there are primarily two parameters to choose: the radius h_n of the local circular neighborhood $B(x, y; h_n)$, and the tuning parameter κ for deciding whether a clustering is necessary or not in a local circular neighborhood. However, it is very difficult to provide a direct formula for selecting the parameters. Their reasonable values in a finite sample practical application depend on the noise level, image dimension, image complexity and other factors. The corollary in Section 4.4 provides a bound for the selection of κ. However, we hereby provide a practical guideline for the selection of the procedure parameters in finite sample applications.

Notably, the tuning parameter κ makes the proposed method more flexible. Based on the practical context, we may choose an appropriate value of κ to obtain meaningful performance. For more discussions related to the flexibility of the tuning parameter, readers are referred to Section 4.5.2.

Selection of the bandwidth parameter h_n: The selection of h_n is an important task. In this chapter, we choose h_n by the leave-one-out cross-validation (CV) technique. We follow the suggestion by Kang et al. (2018) and use the DRIP package in the CRAN R repository [Kang (2023)]. Additionally, larger values of h_n provide better performance for detecting changes, while smaller values are computationally easy. Based on the CV procedure, we suggest choosing $h_n \in [\frac{2.5}{n}, \frac{4}{n}]$. Readers are referred to use the function: *surfaceCluster_ bandwidth()* in the DRIP package for the selection of h_n. See Figure 4.11 in the Section 4.7 for the simulation results on the cross-validation.

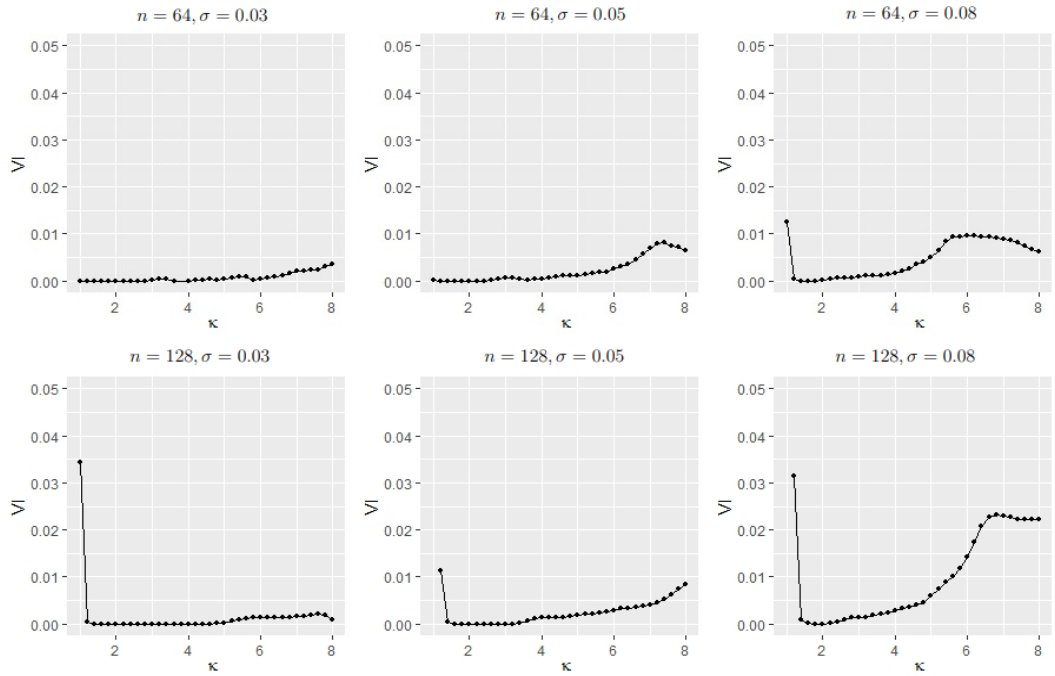


Figure 4.4: Selection of κ for images with various dimensions using the bootstrap procedure: (a) the plots in the upper panel are for the simulated image with dimension 64×64, and (b) the plots in the lower panel are for the simulated image with dimension 128×128.

Selection of the tuning parameter κ: As the proposed method is based on the equality of the local clusters, it is very important to choose the value of κ appropriately in a data-driven manner. Note that larger values of κ are always better for detecting big changes in the images but there is a possibility of not detecting small changes of image intensity values. In this experiment, for each κ, we generate $B = 50$ bootstrap samples by adding residuals with the smoothed image and calculate the VI distance between the clustering of the bootstrapped sample and the smoothed image. This is similar to the procedure described in Section 4.2.4. A reasonable value of κ is obtained by minimizing the average VI distance. In Figure 4.4, we plot various values of κ in the direction of the X-axis and the average VI distance calculated from the bootstrapped samples along the Y-axis. For the simulated images in Section 4.5, we plot three curves for three different noise levels with a fixed bandwidth parameter $h_n = 3$. In the numerical section, we present the performances of the proposed method for various values of κ for images with dimensions 64×64 and 128×128. Considering all images of different noise levels from Figure 4.4, we suggest $\kappa \in [1.2, 2.5]$ for 64×64 simulated images and $\kappa \in [1.5, 2.0]$ for images with dimension 128×128. In the Section 4.7, we have discussed another data-driven method for selecting the tuning parameter based on the number of circular neighborhoods with clustering. See the Section 4.7 for more details.

4.4 Statistical Properties

In this section, we investigate certain statistical properties of the proposed image comparison procedure. In this chapter, a point (x, y) is called a *singular* point if one of the following conditions is satisfied. (i) There exists $\rho > 0$ such that, for any $0 < \bar{\rho} < \rho$, the circular neighborhood of (x, y) with radius $\bar{\rho}$ contains more than two connected regions. (ii) The jump size at (x, y) is zero, that the point (x, y) is one of those $\{(x_k^*, y_k^*), k = 1, 2, \ldots, K^*\}$ as defined in A.3. Moreover, define

$\Omega_\epsilon = [\epsilon, 1-\epsilon] \times [\epsilon, 1-\epsilon]$,
$E_\epsilon = \{(x, y) : (x, y) \in \Omega, D((x, y), (x', y')) \leq \epsilon \text{ for some } (x', y') \in E\}$,
$S_\epsilon = \{(x, y) : (x, y) \in \Omega, D((x, y), (x', y')) \leq \epsilon \text{ for some singular } (x', y') \in E\}$,
$\Omega_{\bar{E},\epsilon} = \Omega \backslash E_\epsilon$ and $\Omega_{\bar{S},\epsilon} = \Omega \backslash S_\epsilon$,

where ϵ is a small positive constant, D denotes Euclidean distance and E defines the set of points on JLCs. The proposed method has two major steps that we justify theoretically. We have the following two propositions and the sketches of their proofs are provided in the Appendix A.3.

Proposition 4.4.1. *Assume that f has continuous first-order derivative over $(0,1)\times(0,1)$ except on the JLCs, its first-order derivatives have one-sided limits at nonsingular points of the JLCs, $\{\varepsilon_{ij}\}$ are iid with mean 0 and finite variance σ^2, where $\sigma^2 > 0$, $h_n = o(1)$, $1/nh_n = o(1)$. Then, we have*

(i) if $(x, y) \in \Omega_{\bar{E},\epsilon}$, then $S(x, y, h_n) \leq \kappa\widehat{\sigma}$ a.s. for any fixed value of $\kappa \geq 1$.

(ii) On the other hand, if $(x, y) \in E_{h_n}$, and further assume that $\frac{n_1}{n_2} = O(1)$ and $\frac{n_2}{n_1} = O(1)$, where n_1 and n_2 are respectively the number of pixel coordinates in two image segments inside the circular neighborhood $B(x, y; h_n)$, then $S(x, y, h_n) > \kappa\widehat{\sigma}$ a.s. as $n \to \infty$.

Remark 4.4.2. *The above proposition theoretically ensures effectiveness of the proposed clustering rule in both absence and presence of the JLCs in a circular neighborhood $B(x, y; h_n)$ for an arbitrary point (x, y). In the asymptotic sense, to detect a difference in the images due to any JLC of any jump size, $\kappa = 1$ is a reasonable choice. However, with a suitable value of κ, the proposed method is capable to ignore differences in the images due to JLCs up to a certain jump size. Such a flexibility of the proposed method is useful in many practical situations. See Section 4.6 for a related discussion. In finite sample applications, the choice of κ, however, should depend on both the noise level and image resolution.*

Corollary 4.4.3. *Suppose $(x, y) \in E_{h_n}$ and ζ is a jump size that one wishes to ignore in the circular neighborhood $B(x, y; h_n)$. Then, the proposed algorithm decides not to cluster in $B(x, y; h_n)$ almost surely, if $\kappa \geq \sqrt{(1 + \zeta^2/\sigma^2)}$*

The above corollary gives an idea about the value of the tuning parameter κ based on situations. It is evident that for a fixed value of σ, to ignore a large jump size, we need a large value of κ. Moreover, for a fixed jump size with large value of σ, the proposed algorithm decides to cluster with small value of κ. As a result, there would be a large number of false clustering and hence performance of the algorithm would become worse. Therefore, it is difficult to handle small jump size with high noise level. This is natural and holds true for any clustering algorithm. Consequently, for large σ, we need to focus on large jump size, and to prevent the effects of false clustering, we choose large κ.

Our next proposition is regarding the effectiveness of the proposed testing procedure based on the VI metric.

Proposition 4.4.4. *Let the maximum mis-classification probability by local pixel clustering be δ. Then, under the assumptions in Proposition 1,*

$$T_{VI}^{H_0} = O(\delta) \ a.s.,$$

$$T_{VI}^{H_1} = O(\delta) + c \ a.s.,$$

where $T_{VI}^{H_0}$ and $T_{VI}^{H_1}$ are the values of the proposed test statistic under the null and alternative hypothesis, respectively, and $c > 0$.

Remark 4.4.5. *From the above proposition, it is clear that as long as the local clustering is good, i.e., the mis-classification probability is very small, the proposed test is capable of separating the distributions of $T_{VI}^{H_1}$ and $T_{VI}^{H_0}$.*

The sketches of the proofs of Propositions 4.4.1 and 4.4.4 are provided in Appendix A.3.

4.5 Numerical Studies

In this section, we numerically evaluate the performance of our proposed image comparison method. To assess the performance of the proposed method, we perform comparative studies on various simulated images. In the literature, plenty of the methods are based on intensity values and therefore, they are very sensitive to noise and unimportant changes in the image contrast. However, in satellite imaging, medical imaging, etc., there are

many situations where small changes in the intensity values in the background are not considered as actual meaningful changes in the images. See Figure 4.6 for a toy example of a smooth change in the background and Figure 4.1 for an unimportant change in JLCs. Our proposed method is based on the important JLCs as primary image features and hence it can raise signal for the changes in JLCs of interest. Therefore, it has wide applications in satellite imaging studies and medical imaging. To assesses the numerical performance of the proposed method, we hereby consider the following competing methods:

- Nonparametric comparison of regression surface proposed by Wang and Ye (2010).
- Image comparison based on continuity and discontinuity region, proposed by Feng and Qiu (2018).

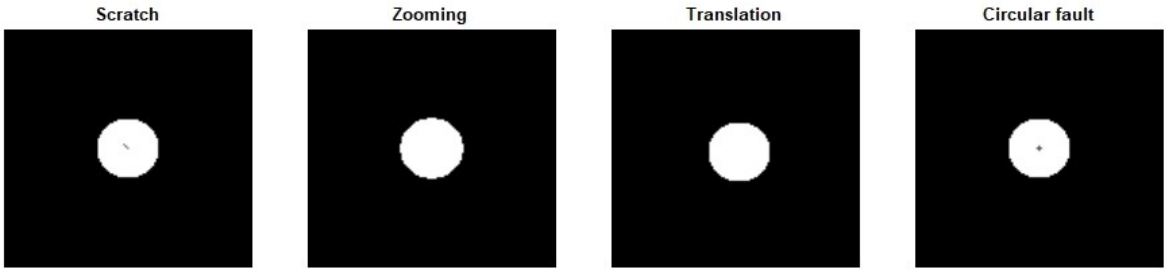


Figure 4.5: Altered images without noise that we consider to evaluate the performance of the proposed method.

4.5.1 Brief discussions of the competing methods

Based on nonparametric comparison of regression surfaces, the method proposed by Wang and Ye (2010) considers the difference between the estimates of the individual image intensity surfaces by $\mathcal{L}^2$-distance as the test statistic. The test statistic is thus defined as

$$T_N = \frac{1}{n^2} \sum_{i,j=1}^{n} (\widehat{f}_1(x_i, y_j) - \widehat{f}_2(x_i, y_j))^2, \tag{4.5.1}$$

where $\widehat{f}_1(x_i, y_j)$ and $\widehat{f}_2(x_i, y_j)$ are the local estimators for the images. Moreover, the above test statistic is a multi-dimensional generalisation of the test statistic defined in Dette and Neumeyer (2001).

Feng and Qiu (2018) propose a testing procedure for image comparisons based on (i) continuity regions of the images, (ii) regions around the detected edge pixels, and (iii) a combination of both continuity regions and discontinuity regions. Define the sets of detected edges for two observed images by D_1 and D_2, respectively. Also, define $D_k(h_G)$,

$k = 1, 2$, as the set of pixels whose Euclidean distance to D_k is less than or equal to h_G and $h_G > 0$ is a bandwidth. Then, a natural test statistic based on continuity region has been defined as:

$$U_C = \sum_{(x_i,y_j)\in G} (w_{ij}^{(1)} - w_{ij}^{(2)})^2, \tag{4.5.2}$$

where $G = \Omega \setminus (D_1(h_G) \cup D_2(h_G))$. Under the null hypothesis, the value of the statistic would be small and otherwise it would be large. For the test at level-α, U_C rejects H_0 if $U_C > z_\alpha^C$. Similarly, the test statistic depending on detected edges has been defined as

$$U_E = \sum_{(x_i,y_j)\in D_1} (w_{ij}^{(1)} - w_{ij}^{(2)})^2, \tag{4.5.3}$$

and reject the null hypothesis if $U_E > z_\alpha^E$. The value of z_α^C and z_α^E can be obtained from the $(1-\alpha)$-quantile of the empirical null distribution of U_C and U_E, respectively. Finally, the testing procedure based on the combination of the above two test statistic can be expressed as follows:

$$U_{EC}\text{: Reject } H_0 \text{ if } U_C > z_{\frac{\alpha}{2}}^C \text{ or } U_E > z_{\frac{\alpha}{2}}^E,$$

where $z_{\frac{\alpha}{2}}^C$ and $z_{\frac{\alpha}{2}}^E$ are the $(1 - \frac{\alpha}{2})$-quantile of U_C and U_E, respectively, under the null hypothesis.

4.5.2 Simulations

Here we are comparing two images and without any loss of generality, consider the first image (f_1) as the reference image. For the purpose of simulation, we consider the true intensity function of the reference image as

$$f_1(x, y) = \mathbf{I}(\sqrt{(x - 0.5)^2 + (y - 0.5)^2} < 0.125), \tag{4.5.4}$$

where $(x, y) \in [0, 1] \times [0, 1]$ and $\mathbf{I}$ is the indicator function. Under H_0, the true intensity functions of both the images are same *i.e.,* $f_1(x, y) = f_2(x, y)$ and as an alternative hypothesis, we consider the following scenarios:

(a) Scratch: $f_2^{(1)}(x, y) = \begin{cases} f_1(x, y) - 0.7 & \text{if } x = y \text{ and } (x, y) \in [0.48, 0.5] \times [0.48, 0.5] \\ f_1(x, y) & \text{otherwise} \end{cases}$

(b) Zooming: $f_2^{(2)}(x,y) = \begin{cases} 1 & \text{if } \sqrt{(x-0.5)^2+(y-0.5)^2} < 0.133 \\ 0 & \text{otherwise} \end{cases}$

(c) Translation: $f_2^{(3)}(x,y) = \begin{cases} 1 & \text{if } \sqrt{(x-0.51)^2+(y-0.51)^2} < 0.125 \\ 0 & \text{otherwise} \end{cases}$

(d) Circular Fault: $f_2^{(4)}(x,y) = \begin{cases} 0.3 & \text{if } \sqrt{(x-0.5)^2+(y-0.5)^2} < 0.016 \\ f_1(x,y) & \text{otherwise} \end{cases}$

Noise Level		T_{VI} $\kappa = 1.3$	T_{VI} $\kappa = 1.4$	U_E	U_C	U_{EC}	T_N
$\sigma = 0.03$	H_0	0.008	0.002	0.000	0.004	0.000	0.000
	Scratch	1.000	1.000	0.000	0.996	0.790	0.000
	Zooming	1.000	1.000	1.000	0.004	1.000	0.000
	Translation	1.000	1.000	1.000	1.000	1.000	1.000
	Circular Fault	1.000	1.000	0.000	0.928	0.188	0.000
$\sigma = 0.05$	H_0	0.088	0.010	0.000	0.008	0.000	0.000
	Scratch	1.000	1.000	0.000	0.364	0.040	0.000
	Zooming	1.000	1.000	1.000	0.008	1.000	0.000
	Translation	1.000	1.000	1.000	1.000	1.000	1.000
	Circular Fault	1.000	1.000	0.000	0.230	0.014	0.000
$\sigma = 0.08$	H_0	0.112	0.066	0.000	0.008	0.036	0.000
	Scratch	1.000	1.000	0.000	0.000	0.114	0.000
	Zooming	1.000	1.000	1.000	0.000	1.000	0.000
	Translation	1.000	1.000	1.000	0.938	1.000	1.000
	Circular Fault	0.286	0.926	0.000	0.000	0.088	0.000

Table 4.1: Empirical sizes and powers for various images of resolution 64 ×64.

Note that the cases (a)-(d) correspond to the alternative hypothesis. In all cases, (x,y) are generated from a regular grid of values in the design space Ω with a fixed and equally spaced design. Figure 4.5 shows these four alternative images. For the simulation purpose, we consider two different sample sizes 64×64 (i.e., $n = 64$) and 128×128 (i.e., $n = 128$), and also consider $\varepsilon(x,y)$ as a random sample from $N(0, 0.03^2)$, $N(0, 0.05^2)$, and $N(0, 0.08^2)$. To find the bootstrap distribution of the test statistic under H_0, we use the algorithm presented in Section 4.2.4, and set $B = 500$. Moreover, to evaluate the performance of the proposed method in comparison with the competing methods mentioned above, we use the empirical size and power. The simulation results for these image resolutions and noise levels are shown in Table 4.1 and Table 4.2. From Figure 4.4, a reasonable choice of κ are 1.3 and 1.4 for image dimensions 64×64. Similarly, for images

Noise Level		T_{VI} $\kappa = 1.5$	T_{VI} $\kappa = 1.8$	U_E	U_C	U_{EC}	T_N
$\sigma = 0.03$	H_0	0.038	0.000	0.000	0.000	0.004	0.000
	Scratch	1.000	1.000	0.000	0.234	1.000	0.000
	Zooming	1.000	1.000	1.000	0.000	1.000	0.000
	Translation	1.000	1.000	1.000	1.000	1.000	1.000
	Circular Fault	1.000	1.000	0.000	1.000	1.000	0.000
$\sigma = 0.05$	H_0	0.038	0.002	0.000	0.068	0.270	0.000
	Scratch	1.000	1.000	0.000	0.518	0.830	0.000
	Zooming	1.000	1.000	1.000	0.068	1.000	0.000
	Translation	1.000	1.000	1.000	1.000	1.000	1.000
	Circular Fault	1.000	1.000	0.000	0.990	0.980	0.000
$\sigma = 0.08$	H_0	0.036	0.062	0.000	0.112	0.380	0.000
	Scratch	0.998	1.000	0.000	0.278	0.750	0.000
	Zooming	1.000	1.000	1.000	0.112	1.000	0.000
	Translation	1.000	1.000	1.000	1.000	1.000	1.000
	Circular Fault	1.000	1.000	0.000	0578	0.850	0.000

Table 4.2: Empirical sizes and powers for various images of resolution 128 ×128.

with dimension 128×128, we have reported the performance with κ equals to 1.5 and 1.8. The average performance of the proposed method based on 500 replicated simulations are shown in Table 4.1 and Table 4.2. From the tables, we can have the following conclusion for the proposed method: (i) For a suitable value of the tuning parameter κ, the empirical sizes for all the tests are quite small. (ii) With a suitable choice of κ, the proposed method is able to detect the changes almost accurately. Regarding two competing methods, we can see that the performance of T_N is poor. Most of the time, it fails to reject the null hypothesis when the alternative hypothesis is correct. T_N is able to correctly detect a change in case of zooming only, because in case of zooming, the change is comparatively larger than the other changes shown in Figure 4.5. Therefore, we find that T_N doe not

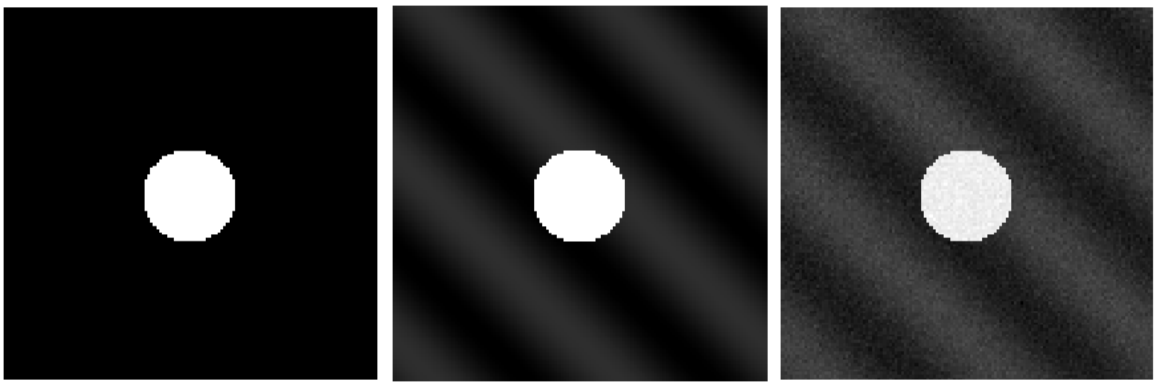


Figure 4.6: True reference image (left), image with smooth changes in the background (middle), and noisy image with smooth changes in the background (right). These images are of dimension 128×128.

perform well for detecting small changes. The method based on testing equality in the continuity and discontinuity regions performs much better than T_N. Similar to T_N, a test based on U_E could not detect the changes for scratch and circular faults. Performance with U_C is comparatively better in those situations. However, at high noise levels, it fails to detect changes accurately. In case of U_{EC}, for 128×128 images with large noise, the empirical size is quite large compared to the actual size. Therefore, it needs to be carefully used in practical situations. The comparison of the proposed algorithm is quite illustrative. It outperforms the competing methods in nearly all situations mentioned above. When the noise level is comparatively higher, in cases of scratches and circular faults, the performance of this method is not up to the mark but much better than the existing methods. Therefore, from Table 4.1 and 4.2, it is quite evident that the proposed method is a winner in almost all cases.

σ	$\kappa = 1.20$	$\kappa = 1.40$	$\kappa = 1.60$
0.03	0.806	0.472	0.032
0.05	0.434	0.234	0.074
0.08	0.158	0.070	0.050

Table 4.3: Empirical powers for various values of κ for the image in Figure 4.6.

Now, we present an interesting situation to ensure the flexibility and effectiveness of our proposed method. In real life applications, there are many situations where a small change in the intensity in the background does not imply actual meaningful change in the image object. As discussed earlier, one major advantage of the proposed method is that it can address such situations depending on the purpose the application. To describe this scenario, consider an image with a small smooth change in the background. True intensity function of the image in the middle panel of Figure 4.6 is given by:

$$g(x,y) = \begin{cases} 1 & \text{if } \sqrt{(x-0.5)^2 + (y-0.5)^2} < 0.133 \\ \frac{1}{10}\sin(9.6(x+y)) & \text{otherwise} \end{cases}$$

From the above functional form of the image surface, it is clear that the image object and the JLCs remain unchanged in across all panels of Figure 4.6. Similar situations are common in many real life applications. As the image object remains the same, the test of image comparison should not reject the null hypothesis. However, most of the existing methods are very sensitive to these changes. From Table 4.3, it is clear that the proposed algorithm ignores the above smooth insignificant change with a suitable choice of the tuning parameter (for $\kappa = 1.6$). Therefore, the choice of κ makes the proposed method flexible in handling many practical situations.

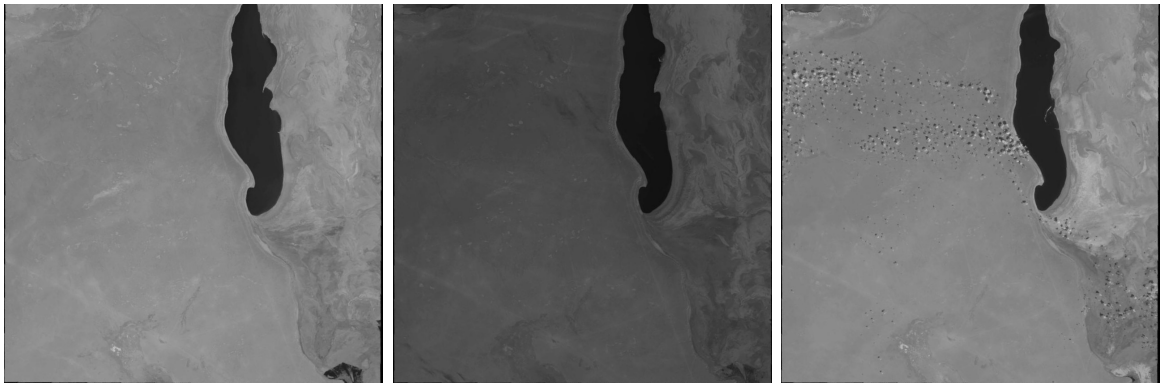


Figure 4.7: Three satellite images of the Aral sea area taken in 2016 (left), 2018 (middle), and 2022 (right).

4.6 Real Data Application

In this section, we demonstrate the performance of the proposed testing procedure for image comparison (4.2.7) on real image data from the LANDSAT project (https://earthexplorer.usgs.gov/). In the Aral sea example in Figure 1.1, the area of the sea changes gradually over time. It was once the fourth largest lake in the world and has shrunk to more than half of its former size. As a direct consequence of the shrinkage of the lake, fisheries and communities that depend on them have collapsed. Also, it has a wide effect on damaging the local ecosystems and sustainable cultivation. So for monitoring the sea area over time, comparing images of Aral sea at different time points is an important task. In this example, we are mainly interested in the area of Aral sea. The images at different time points are shown in Figure 4.7. Here, it is evident that there

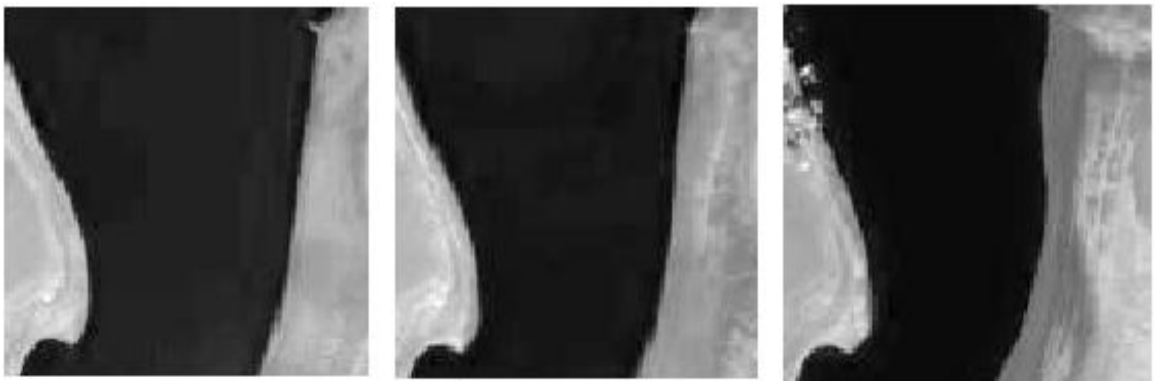


Figure 4.8: Cropped satellite images of Aral sea taken in 2016 (left), 2018 (middle), and 2022 (right).

is a change in the contrast for the image taken in 2016 and the image taken in 2018. As the proposed method is feature based, it is capable of ignoring this meaningless change. Also, for the image in 2022, some insignificant spots are present in the background. There is a possibility that the proposed method may detect a change only due to a change in the

background. However, the proposed method is flexible enough to ignore these background spots. For this reason, we choose a large value of κ. To make the comparison reliable, in the pre-processing step we register the images properly and for computational simplicity we cropped the image properly. All cropped images have dimension 126×126. See Figure 4.8 for the cropped images. To find the empirical distribution of the test statistic, we first calculate the residuals from the jump preserving estimate and generate bootstrapped image samples by adding bootstrapped residuals to it.

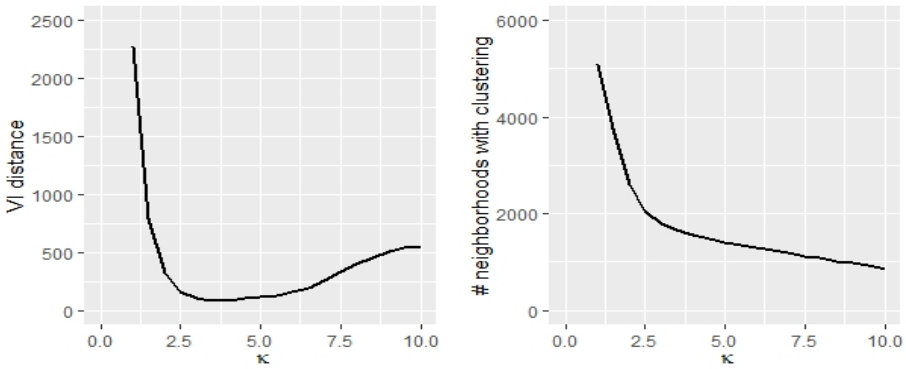


Figure 4.9: Selection of the tuning parameter κ.

For the comparison purpose, we test the equality of images from 2016, 2018 and 2022. Estimated noise ($\widehat{\sigma}$) from the residuals are $0.011, 0.009$, and 0.017, respectively. From cross-validation, the suitable choice for h_n is $\frac{3}{126}$. As in this applications we are interested in the large change (change of the area of Aral sea) it is preferable to choose a κ that ignores jumps of size at least three times $\widehat{\sigma}$. Therefore, from the corollary in Section 4.4, it is preferable to consider a κ around 3. Also, from the bootstrap procedure described in Section 4.2.4, a suitable choice of κ lies in $[3, 5]$. See Figure 4.9, for the tuning parameter selection using bootstrap procedure for the image of Aral sea in 2018. However, the image in 2022 contains background spots that should be avoided in the comparison. To ignore avoidable JLCs the background, we recommend to choose $\kappa \in [9, 10]$. The readers are referred to Section 4.7 for the selection of suitable κ in this example. For testing "2016 vs 2018", the estimated p-value with 100 replications and $\kappa = 4$ is 0.0012. For testing "2016 vs 2022" after ignoring the unimportant JLCs with $\kappa = 10$, the estimated p-value is 0.0026. The changes of the major regions of Aral sea are presented in Figure 4.10. Images in Figure 4.10 validate the performance of the proposed method as well. Therefore, from the p-values mentioned above, it is evident that the proposed method is capable of detecting changes very efficiently.

Figure 4.10: Changes from the image in 2016 to 2018 (left), 2016 to 2022 (middle), and 2018 to 2022 (right).

4.7 More Details on The Selection of Procedure Parameters

Selection of the bandwidth parameter: In Figure 4.11, we plot the CV scores along the Y-axis and various bandwidth parameter values along the X-axis. From the CV scores, the preferable choice of h_n is $[\frac{2.5}{n}, \frac{4}{n}]$. For calculating the CV scores, we use the function: *surfaceCluster_ bandwidth()* in the R package DRIP.

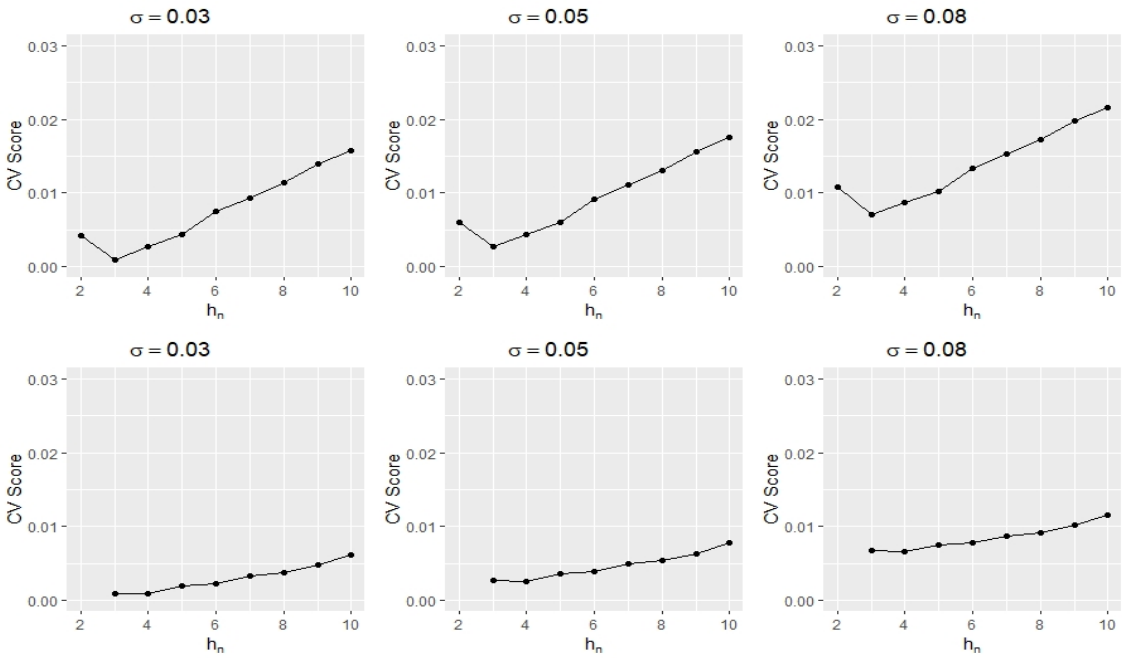


Figure 4.11: Plot of the CV scores for the simulated image with various noise levels: (a) The plots in the upper panel are for the image with dimension 64×64, and (b) the plots in the lower panel are for the image with dimension 128×128.

Selection of the tuning parameter κ : Section 4.3 provides a guideline for selecting κ by minimizing the average VI metric of bootstrapped sample images. Here we present an

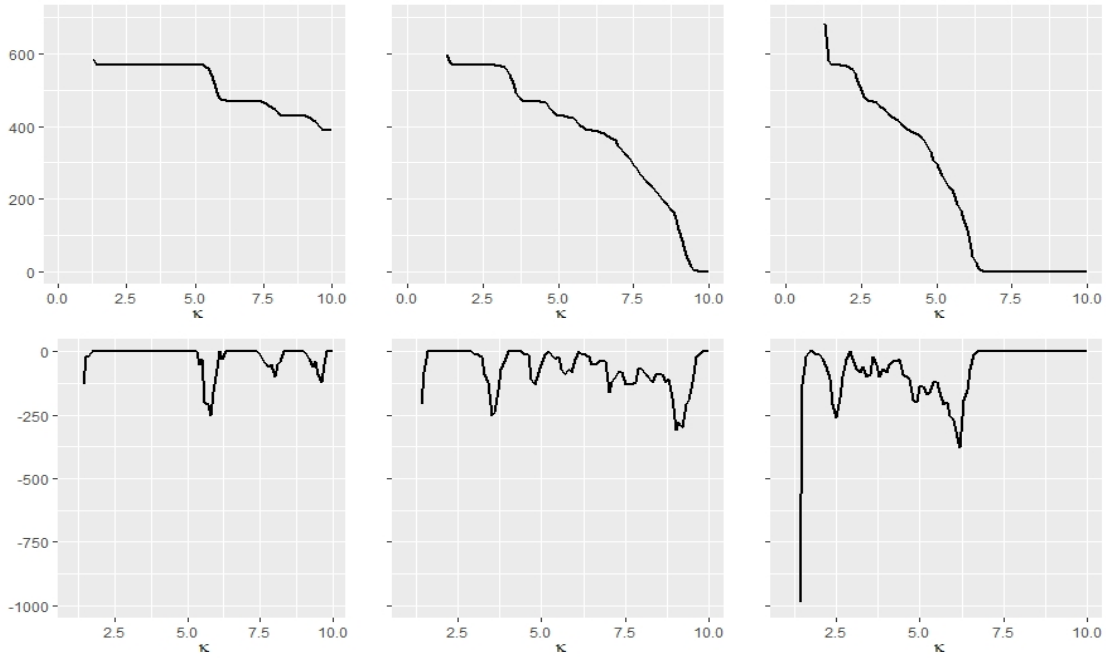


Figure 4.12: Selection of κ based on the number of neighborhoods undergoing clustering: (a) In the sub-figures of the upper panel, we plot the number of clustering along the Y-axis and κ along the X-axis, (b) in the sub-figures of the lower panel, we plot the slope of the corresponding curves in the upper panel.

alternative way of choosing κ based on the number of circular neighborhoods where the proposed algorithm performs local clustering. In the first row of Figure 4.12, we plot the number of neighborhoods undergoing local clustering along the Y-axis and various values of κ along the X-axis. It is preferable to choose a κ where the the curve is relatively flat, i.e., the slope is close to zero.

Demonstration of the selection of κ for a real image: To get rid of the background disturbances, we focus on two different parts of the same image of Aral sea in 2022 (two panels in Figure 4.13). First we aim to get an idea about the jump size that we wish to ignore. The image in the left panel of Figure 4.13 contains background disturbances only. However, they are JLCs, but we wish to disregard them for the purpose of image comparison with another image of Aral sea collected in another year. Table 4.4 presents the number of circular neighborhoods undergoing local clustering in the background region (viz., the left panel of Figure 4.13) with various values of κ, and we recommend the values of κ for which no local clustering is performed. Conversely, the image on the right panel of Figure 4.13 contains no background disturbance. Therefore, a suitable choice of the tuning parameter is a value that will do no clustering in the left image of Figure 4.13, and maximum clustering in the right image of Figure 4.13 In Section 4.6, we demonstrate the performance of the proposed algorithm with various values of κ. However, from Table 4.4, it is advisable to consider $\kappa = 9$ or 10 for the image of the Aral Sea area in 2022 that

contains both the disturbances and a part of Aral sea.

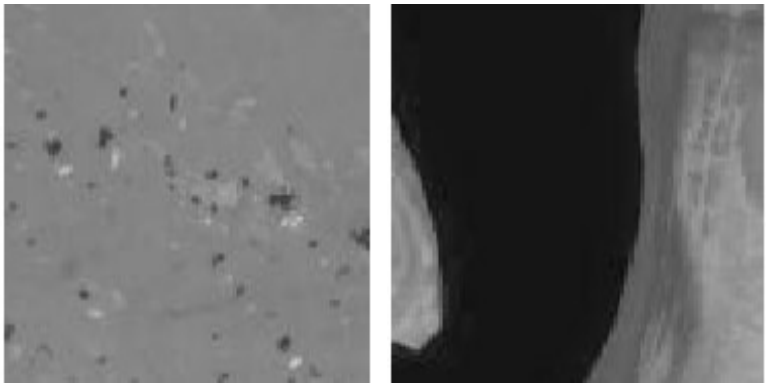


Figure 4.13: Two different regions of the Aral sea area in 2022. Only background disturbances in the left panel, and a part of Aral sea without any disturbances in the right panel.

κ	Ignorable JLCs	Important JLCs
2.00	2475	2392
3.00	1500	1384
5.00	590	945
8.00	8	779
9.00	0	720
10.00	0	674
12.00	0	575
15.00	0	393

Table 4.4: The number of neighborhoods undergoing local clustering for various values of κ.

4.8 Concluding Remark

This chapter proposes a feature-based image comparison method based on local pixel clustering. Most of the existing methods are very sensitive to small changes in the background that are often unimportant in practical applications, and therefore, such methods are sometimes unreliable in real life scenarios. However, our method has a mechanism to detect the actual change over unwanted small changes in the background. In that sense, it is more flexible and has more reliable performance in many real life applications. Numerical comparisons with two state-of-the-art methods ensure that it has several merits. There are several future directions for our current research work. Monitoring a sequence of images over time by the clustering based metrics is a very simple and natural extension. In many image applications, the images could contain spatial blur [Kang et al. (2018)]

in addition to the random noise. Although our clustering based comparison method has certain ability to deal with blurring effects, special attention should be given for image comparisons in presence of spatial blur. Most image comparison methods consider image registration as a pre-processing step. However, a method that can handle image misalignment by its construction would be more suitable for real life applications.

Chapter 5

Shape and Size Monitoring in Presence of Rigid-body Image Transformation

5.1 Introduction

Recently, satellite images are capturing considerable attention for monitoring the earth's surface over time. The LANDSAT Project, which began in 1972, is a well-known initiative in which the USGS and NASA launched nine satellites till date, to collect images of the earth's surface over time. Figure 5.1 shows four satellite images of the Salton sea area located at the southern end of the U.S. state of California. By comparing the sequence

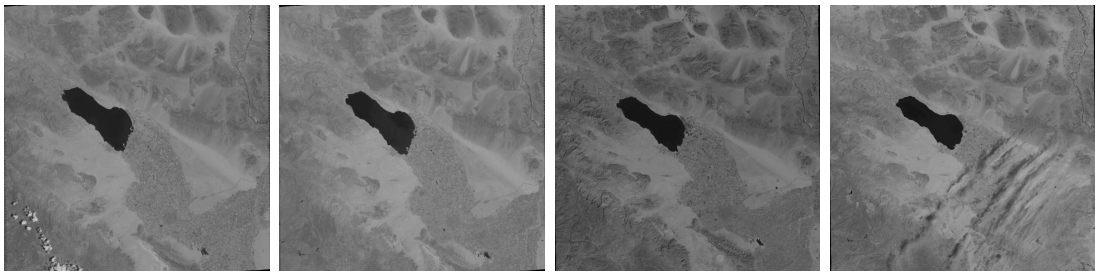


Figure 5.1: The satellite images of the Salton sea area captured in 2004, 2010, 2018 and 2023, respectively.

of images, one can monitor the earth's surface and natural resources around this region. For example, it is quite evident from Figure 5.1 that the upper part of the lake boundary moved downward drastically in the 19 years of time span. Therefore, it has become an important to monitor shape and size of this type of satellite images.

However, monitoring this type of image data is indeed a challenging problem because of two reasons. Firstly, discontinuous image surface contain noise. Secondly, and more importantly, the real images of the same object captured by different sensors or at different time points are often not geometrically aligned. To demonstrate the role of image registration in the context of image comparison and image monitoring, consider the following example in Figure 5.2. The first two images from the left are the satellite image of the Salton sea area in 2005 and 2008, respectively. The third one from the left is the pixel-wise difference between these two images without image registration, and the image

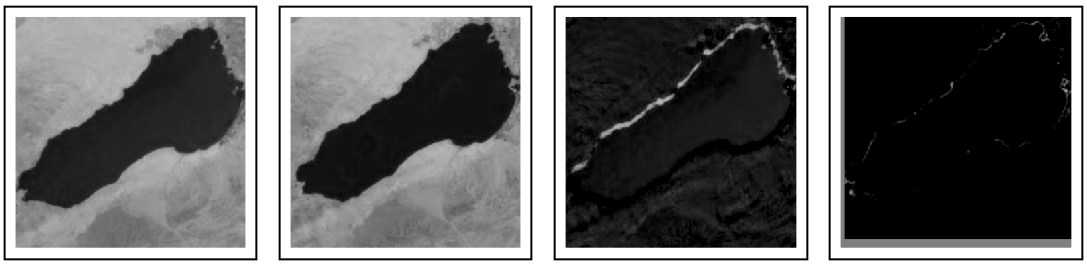


Figure 5.2: (a) The Salton sea area in 2005, (b) the Salton sea area in 2008, (c) pixel-wise difference without image registration, and (d) pixel-wise difference after image registration.

on the right is the pixel-wise difference between the two images after image registration. From Figure 5.2, it is evident that image monitoring without aligning images properly leads to significant false conclusions. In case of online image monitoring, performing image registration at the arrival of each new image is computationally extensive as well. This chapter aims to overcome the issue of image registration while performing both image comparison and image monitoring.

In literature, methods related to image monitoring assume proper geometrical alignment of the images. Moreover, they are very sensitive to minor and unimportant alteration of the image intensity values. Recently, Feng and Qiu (2018); Roy and Mukherjee (2024a,b) propose methods for image comparison and monitoring based on jump location curves (JLCs). These methods are capable of disregarding small unimportant changes in the image intensities. However, it is worth mentioning that they still have to use an appropriate image registration technique to geometrically align the images beforehand.

In this chapter, we propose an image comparison and monitoring procedure to determine whether the shape and size of an image object at different time points are unchanged or not. The novelty of the proposed method is that it does not require image registration during the pre-processing phase. To the best of our knowledge, no image comparison or monitoring technique in the literature can take the geometric misalignment into consideration by its construction, not employing an image registration procedure during pre-processing. Moreover, the outcomes of the image registration step that the existing

methods require, are often not error-free. Another critical point to note here is that if the shape or size of an image object is actually different in two images, the rigid-body image registration is no longer a valid problem, interpretation of the registered image is not obvious. Other major benefits of using the proposed method are: it is computationally less expensive, and more accurate. Additionally, the proposed algorithm does not directly depend on the image intensity values, and therefore capable of ignoring small unimportant changes in the image intensity values in the foreground or the background of the images. Such situations are very common in satellite imaging. For example, images of the same scene in the presence and absence of clouds. The proposed image comparison and monitoring procedure has the following steps. We first detect the edges of the image object by using a state-of-the-art jump detection algorithm and estimate the centroid of the image object by calculating the sample mean of the coordinates of the detected edge pixels. In each image, we estimate the distribution of the radial distances of the edge pixels from the centroid by a suitable estimator, and develop a test statistic based on Cramér-von Mises criterion. Finally, we construct a CUSUM control chart for online monitoring of the sequential images. For better demonstration, refer to the toy image in Figure 5.3. In this example, we consider two image objects of different shapes: a circle and an ellipse.

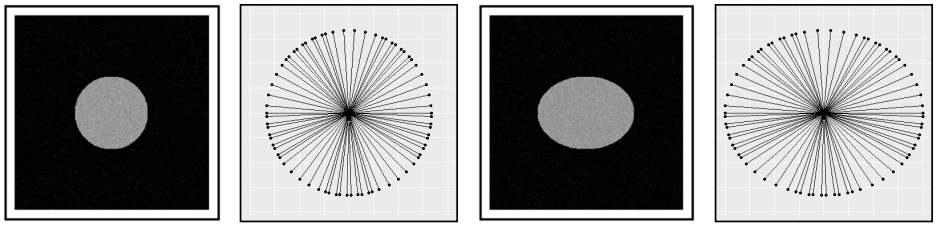


Figure 5.3: Graphical demonstration of shape & size comparison based on radial distances.

The 2-nd and 4-th image of Figure 5.3 show the distances from the estimated centroids to the detected edge points on the boundaries of the circle and ellipse, respectively. We call such distances as radial distances. Figure 5.4 shows the distribution functions of the radial distances corresponding to the circle and ellipse.

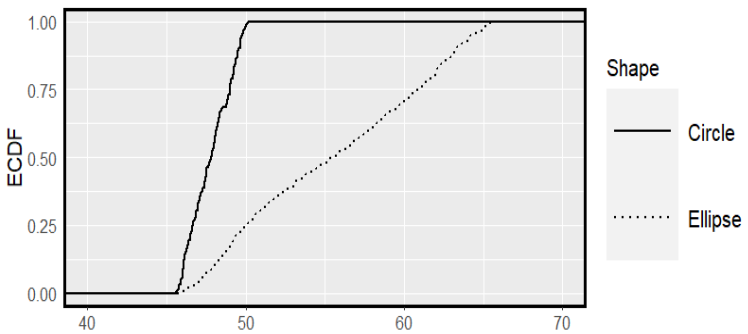


Figure 5.4: The distribution functions of the radial distances corresponding to the circle and ellipse.

5.2 Proposed Methodology

Essentially, our goal in this chapter is to propose an image monitoring procedure that is invariant under rigid-body transformation of the image object. Therefore, the proposed algorithm is capable of detecting a change in the shape and size of the image object quite accurately while accommodating the rigid-body transformation of the image objects, such as translation and rotation. Although the proposed method can be applied to various types of images, we consider only the 2-D monochrome images in this chapter. As we are mainly interested in monitoring the shape and size of one image object, we assume that the image object has a closed shape. In this section, we describe the proposed algorithm in a step by step manner.

5.2.1 An LLK-based method for jump location curve estimation

In the JRA literature, a 2-D grayscale image can be expressed by the following nonparametric regression model [Qiu (2009)]:

$$w_{ij} = f(x_i, y_j) + \varepsilon_{ij}\ , \quad \text{for} \quad i, j = 1, 2, \ldots, n, \tag{5.2.1}$$

where $\{(x_i, y_j) = (i/n, j/n) : i, j = 1, 2, ..., n\}$ are equally spaced pixel coordinates in the design space $\Omega = [0, 1] \times [0, 1]$, w_{ij} is the observed intensity value at the (i, j)-th pixel, $N = n^2$ is the sample size and ε_{ij} are independent and identically distributed (i.i.d.) random errors with mean 0 and variance $\sigma^2 > 0$.

As discussed earlier, a key step for the proposed method is to estimate the boundary of an image object with reasonable accuracy. Theoretically, any edge detection algorithm [Canny (1986); Sun and Qiu (2007)] can be used to automatically estimate the boundary curves or the JLCs. In this chapter, we use an edge detection algorithm based on Local Linear Kernel (LLK) smoothing. Now, at a given pixel (x, y), consider a circular neighborhood $B(x, y; h_n) = \{(u, v) : (u, v) \in \Omega, \sqrt{(u - x)^2 + (v - y)^2} \leq h_n\}$, where $h_n > 0$ is a bandwidth parameter. Then, the LLK smoothing procedure can be accomplished by fitting a local plane in $B(x, y; h_n)$. To do so, consider the following optimization problem:

$$\min_{a,b,c} \sum_{(x_i,y_j)\in B(x,y;h_n)} \Big[w_{ij} - \{a + b(x_i - x) + c(y_j - y)\}\Big]^2 K_{ij}, \tag{5.2.2}$$

where $K_{ij} = K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)$ and K is a radially symmetric bivariate density kernel with support $\{(x, y) : x^2 + y^2 \leq 1\}$. The solution of the above optimization problem is denoted as $(\widehat{a}, \widehat{b}, \widehat{c})$. Note that, $\widehat{a}(x, y)$ is the conventional LLK estimate of

$f(x,y)$, and $(\widehat{b}(x,y),\widehat{c}(x,y))'$ is the LLK estimator of the gradient direction $\boldsymbol{\beta}(x,y) = (f'_x(x,y), f'_y(x,y))'$. Now to detect the jump points, we divide the circular neighborhood $B(x,y;h_n)$ into two parts, denoted as $\mathcal{B}_1(x,y)$ and $\mathcal{B}_2(x,y)$, along the perpendicular direction of the estimated gradient. Using each of the regions $\mathcal{B}_1(x,y)$ and $\mathcal{B}_2(x,y)$, we estimate $f(x,y)$ by LLK method and call them $\widehat{f}_{LLK,\mathcal{B}_1}$ and $\widehat{f}_{LLK,\mathcal{B}_2}$, respectively. A large value of $|\widehat{f}_{LLK,\mathcal{B}_1} - \widehat{f}_{LLK,\mathcal{B}_1}|$ indicates that (x,y) can be a possible jump point. Let us define,

$$LLK(x,y) = \frac{\widehat{f}_{LLK,\mathcal{B}_1} - \widehat{f}_{LLK,\mathcal{B}_2}}{\sqrt{\frac{\sum_{(x_i,y_j)\in\mathcal{B}_1} K_{ij}^2}{[\sum_{(x_i,y_j)\in\mathcal{B}_1} K_{ij}]^2} + \frac{\sum_{(x_i,y_j)\in\mathcal{B}_2} K_{ij}^2}{[\sum_{(x_i,y_j)\in\mathcal{B}_2} K_{ij}]^2}}},$$

then it can be shown that [Kang and Qiu (2014)]:
(i) If f is continuous around (x,y), then $LLK(x,y) \xrightarrow{d} N(0,1)$.
(ii) If f has discontinuities around (x,y), then $LLK(x,y) \xrightarrow{p} \infty$.
Therefore, a design point (x,y) is flagged as a detected jump point if

$$|LLK(x,y)| > \widehat{\sigma} Z_{1-\alpha_n},$$

where $Z_{1-\alpha_n}$, is the $(1-\alpha_n)$ quantile of $N(0,1)$, and $\widehat{\sigma}$ is an estimate of unknown σ. In practice, σ^2 can be estimated by the residual sum of squares divided by the total number of pixels. Then the set of edge pixels is defined as

$$E = \{(x,y) : |LLK(x,y)| > \widehat{\sigma} Z_{1-\alpha_n}\}.$$

Readers are referred to [Qiu (2004)] for more details about the jump detection criterion.

Selection of the bandwidth parameter h_n: Determining the bandwidth parameter h_n is pivotal. In this study, we choose h_n through leave-one-out cross validation (CV) procedure. Based on our numerical experience and CV score, we suggest choosing $h_n \in [\frac{2}{n}, \frac{3}{n}]$. In Section 5.2 (refer to Figure 5.7), we illustrate the CV scores for different values of h_n for the simulated images and identify the optimal bandwidth $(\widetilde{h}_n)$ as the one that minimizes the CV score.

Remarks: Any edge detection technique with reasonable performance can be effectively applied to the proposed methodology. In medical imaging applications, such as tumor detection, it is sometimes necessary to disregard vague boundaries in order to accurately identify the tumor. For such applications, deep learning based image segmentation and edge detection methods are particularly beneficial. Siddique et al. (2021) provide recent and advanced edge detection and image segmentation procedure.

5.2.2 Shape and size comparison

The pivotal step for any image monitoring procedure is image comparison. In this chapter, we propose a test based on the distribution of the distances of all points on the boundary of the image object from its centroid, for comparing the shape and size of the image object in two different images in presence of a rigid-body transformation. In literature, most of the image comparison methods are based on either image intensities or selected image features. However, they are not invariant to rotation and translation. Therefore, to make a reliable image comparison, these existing methods need a separate image registration step as pre-processing. Hence, it is important to construct a feature-based image comparison test that is invariant under rotation and translation. In this chapter, we consider the distances from centroid to all points on the boundary of the image object as our primary feature and construct a test based on that. It can be easily shown that the distribution of the distances from centroid to all points on the boundary curve of the image object is invariant under rotation and translation. See Section 5.3 for more details. We define the distances from the centroid of the image object to the points on the boundary of that image object as radial distances. Therefore, a suitable test of hypotheses that we can use is the following:

$$H_0 : F_1 = F_2 \text{ vs. } H_1 : F_1 \neq F_2, \tag{5.2.3}$$

where F_1 and F_2 are the CDFs of the distances from centroid to all points on the boundary curve of the image object (JLCs) for the two images. To this end, we use jump preserving LLK smoothing on the two images to get the point sets of detected edge pixels corresponding to the boundary of the image object.

Calculation of the empirical distribution function: To determine the empirical distribution, we need to first calculate the jump/edge preserving surface estimates [Qiu (2009)] of the true image intensity functions for the two images, using LLK smoothing. Let the surface estimates for the two images be $\widehat{f}_1$ and $\widehat{f}_2$, respectively. Then the boundary curve or JLC of the image object can be approximated by the point-sets $\widehat{E}_1$ and $\widehat{E}_2$. The centroid of the image object is then estimated by $\widehat{\mathcal{C}}_k = \frac{\sum_{i,j=1}^{n}(x_i,y_j)\mathbb{I}_{(x_i,y_j)}(\widehat{E}_k)}{|\widehat{E}_k|}$ for the k-th image, where $k = 1, 2$. Here, $\mathbb{I}(.)$ is the indicator function. As the point-set of detected edge pixels converges almost surely to the true JLC as the image dimension increases, the estimated centroid of the image object converges to the true centroid as well. Finally, the empirical distribution is obtained by calculating the Euclidean distance between the estimated centroid to each point in the point-set of detected edge pixels. See Section 5.3, for the theoretical justification.

Construction of the test statistic: Suppose, we denote the empirical CDFs of F_1 and F_2 by $\widetilde{F}_1$ and $\widetilde{F}_2$, respectively. Additionally, $\widetilde{F}_c$ is the empirical distribution function of

the combined samples. Assuming $|\widehat{E}_k| = r_k$ for $k = 1, 2$, the proposed test statistic can be expressed as

$$T_{CVM} = \frac{r_1 r_2}{r_1 + r_2} \int_{-\infty}^{\infty} [\widetilde{F}_1(x) - \widetilde{F}_2(x)]^2 d\widetilde{F}_c(x). \tag{5.2.4}$$

Note that the test statistic is similar to the classical Cramér-von Mises statistic. It calculates the sum of squared differences between the two estimated CDFs, evaluated at each sample points of the combined samples. Readers are referred to [Curry et al. (2019); Anderson (1962); Rosenblatt (1952)] for an extensive discussion regarding the Cramér-von Mises statistic.

Determination of critical value for the proposed test: It is a challenging task to theoretically find the asymptotic null distribution of the proposed statistic. In this chapter, we determine the cut-off value of the test statistic by the bootstrapping technique as described below.

1. Under the null hypothesis, the shape and size of the image object are the same after accommodating the fact that rigid-body transformation of the image object may be present. Therefore, to reflect the null hypothesis, we apply rigid-body transformation (i.e., rotation and translation) on the observed image object, and create a pool of image samples, keeping the shape and size of the image object unchanged.

2. Select two images randomly from the pool with replacement, and calculate the value of the proposed test statistic with the two bootstrapped images.

3. Repeat the previous step B times, and use these values of the test statistic to obtain the empirical distribution of the test statistic.

The empirical distribution of under H_0 can be calculated using the above bootstrap procedure. The null hypothesis is rejected for a level-α test if T_{CVM} is greater than the corresponding $(1-\alpha)$-th sample quantile of the bootstrap distribution of T_{CVM}. A theoretical justification for the proposed bootstrap algorithm is outlined in Appendix A.4.

5.2.3 Phase II online monitoring

For the purpose of image comparison, we need to know the true distribution F_0 of radial distances of the image object. Therefore, for Phase-II process monitoring, the true image surface, denoted as $f^{(0)}(x, y)$, should ideally be known. In practice, it is typically unknown, and needs to be estimated from the in-control data. Assume that there are m observed images in Phase-I of image monitoring. These images may or may not be geometrically

aligned, and they are referred as in-control data/images. The true distribution of the radial distances can be reasonably estimated by using all in-control images in Phase-I stage, and we denote the empirical distribution of the radial distances of the combined sample by $\widetilde{F}_0$.

In Phase-II step, the ℓ-th observed image intensities can be expressed by a 2-D JRA model:

$$Z_{ij\ell} = f_\ell(x_i, y_j) + \varepsilon_{ij\ell}; \quad \text{for} \quad i, j = 1, 2, \ldots, n, \text{ and } \quad \ell = 1, 2, \ldots, \tag{5.2.5}$$

where $Z_{ij\ell}$ is the observed image intensity of the ℓ-th image at (i, j)-th pixel, and other related variables can be described similarly as in (5.2.1). To this end, each Phase-II image intensity function is estimated using the jump preserving local linear smoothing technique [Qiu (2009)]. As the proposed algorithm is invariant under rigid-body transformation, the Phase-II step also does not require the images to be geometrically aligned. Therefore, if the process is in-control at ℓ-th time point, i.e., the shape and size of the image object at that time point is same as in the in-control images, then it is natural that $F_\ell = F_0$, where F_ℓ is the distribution of the radial distances for the image at ℓ-th time point. Then, the overall difference between the distribution of radial distances of the image object at ℓ-th time point, and in-control distribution of the radial distances, i.e., $\widetilde{F}_0$ can be determined by the Cramér-von Mises statistic

$$T_{CVM,\ell} = \frac{r_0 r_\ell}{r_\ell + r_0} \int_{-\infty}^{\infty} [\widetilde{F}_\ell(x) - \widetilde{F}_0(x)]^2 d\widetilde{F}_c(x), \tag{5.2.6}$$

where r_0 is the number of detected edge pixels by combining all in-control samples and r_l is the number of detected edge pixels in $\widehat{f}_\ell(x, y)$, respectively . For online monitoring, we construct the following CUSUM statistic at ℓ-th time point:

$$T_{CUSUM,\ell} = \max\left(0, T_{CUSUM,(\ell-1)} + \frac{T_{CVM,\ell} - \mathbb{E}_{IC}(T_{CVM,\ell})}{\sqrt{Var_{IC}(T_{CVM,\ell})}} - \kappa\right) \text{ for } \ell = 1, 2, \ldots, \tag{5.2.7}$$

where $\kappa \geq 0$ is a pre-specified allowance parameter. To choose the allowance parameter optimally, the readers are referred to [Reynolds (1975); Qiu (2013)]. $\mathbb{E}_{IC}(T_{CVM,\ell})$ and $Var_{IC}(T_{CVM,\ell})$, i.e., the in-control mean and variance of $T_{CVM,\ell}$ have to be estimated from the in-control images. Note that, for $\ell = 0$, $T_{CUSUM,\ell} = 0$. Therefore, the proposed algorithm raises a signal of change at ℓ-th time point if $T_{CUSUM,\ell} > t_0$, where t_0 is the control limit. Traditionally, the control limit is chosen in such a way that the in control "*average run length*" (ARL) of the chart statistic can reach the pre-fixed nominal level ARL_0. In this case, the in-control ARL is the expected time to signal a change under the in-control distribution F_0.

Determination of t_0 using a pre-fixed ARL_0: In this chapter, we determine the control limit t_0 by the bootstrap technique mentioned as follows. Similar to Section 5.2.2, we generate bootstrap images by applying rigid-body transformations on the image object. Based on the bootstrapped images, we calculate the CUSUM statistic using equation (5.2.7). For a given value of t_0, we continue the above process until the control chart based on the CUSUM statistic raises a signal of change. To this end, a sample run is found, and the corresponding run length is calculated. We repeat the above-mentioned steps B times, and estimate the actual ARL_0 by the sample mean of the B run lengths. We increase or decrease the trial value of t_0 based on whether the estimated ARL_0 is less than or more than the pre-determined ARL_0 value. The bisection procedure can be used to get the appropriate value of t_0.

5.3 Statistical Properties

In this section, we discuss a number of statistical properties of the proposed method for shape and size monitoring of an image object described in Section 5.2.2. Let us define the following notations:
$\Omega_{h_n} = [h_n, 1-h_n] \times [h_n, 1-h_n]$,
$E_{h_n} = \{(x,y) : (x,y) \in \Omega, D((x,y),(x',y')) \leq h_n \text{ for some } (x',y') \in E\}$,
$S_{h_n} = \{(x,y) : (x,y) \in \Omega, D((x,y),(x',y')) \leq h_n \text{ for some singular } (x',y') \in E\}$,
$\Omega_{\bar{E},h_n} = \Omega \backslash E_{h_n}$ and $\Omega_{\bar{S},h_n} = \Omega \backslash S_{h_n}$,
where ϵ is a small positive constant, D denotes Euclidean distance, E is the set of points on true JLCs, and $\widehat{E}$, is the set of detected edge pixels. Proposition 5.3.1 shows that the estimated centroid of the image object converges almost surely to the true centroid.

Proposition 5.3.1. *Assume that in each closed subset of $[0,1] \times [0,1]$, the unknown true regression function f has piecewise continuous third order derivative. Additionally, on each boundary curve of the pieces in which the first order derivative of f are continuous, the first-order derivatives of f have uniformly bounded one-sided limits defined in individual pieces. Further assume that, $h_n = o(1)$, $\frac{log(n)}{nh_n^4} = o(1)$; $E|\varepsilon_{11}|^2 < \infty$; the kernel function is a Lipschitz-1 continuous circularly symmetric density function on the support $[0,1] \times [0,1]$ and α_n is chosen such that $\alpha_n = o(1)$, $Z_{1-\alpha_n}/(nh_n) = o(1)$, and $(nh_n^3)/Z_{1-\alpha_n} = o(1)$. Then we have,*

$$\widehat{\mathcal{C}} = \mathcal{C} + O(h_n) \ a.s.,$$

where $\mathcal{C} = (\mathcal{C}_x, \mathcal{C}_y)'$ is the true centroid of the image object and $\widehat{\mathcal{C}}$ is the estimate of it.

A sketch of the proof of the above proposition is provided in Appendix A.4. Using Proposition 5.3.1, we can prove the following results. The details of the proofs are also provided in Appendix A.4.

Corollary 5.3.2. *For every* $(x_i, y_j) \in \widehat{E} \cap \Omega_{\bar{S},h_n}$, *there exists at least one* $(x, y) \in E \cap \Omega_{\bar{S},h_n}$ *such that* $D((x_i, y_j), \widehat{\mathcal{C}}) = D((x, y), \mathcal{C}) + O(h_n)$ *a.s.*

Corollary 5.3.3. *Let* F *be the true CDF of the radial distances, and assume that the first order derivative of* F *is bounded. Then as* $n \to \infty$, $\widetilde{F}(z) \xrightarrow{a.s.} F(z)$, *where* $\widetilde{F}$ *is the estimated CDF of* F.

Corollary 5.3.3 ensures that the estimated distribution function converges almost surely to the true distribution function of the radial distances of the image object.

Now we have the next proposition.

Proposition 5.3.4. *Let* F_1 *and* F_2 *be the CDFs of radial distances corresponding to the image object in the given two images. Assume that the first order derivatives of* F_1 *and* F_2 *are bounded, and that* $\frac{r_1}{r_1+r_2} \to \tau$, *where* $0 < \tau < 1$. *Then,*

$$T_{CVM}(\widetilde{F}_1, \widetilde{F}_2) = \frac{r_1 r_2}{r_1 + r_2} \int_{-\infty}^{\infty} [\widetilde{F}_1(x) - \widetilde{F}_2(x)]^2 d\widetilde{F}_c(x) \xrightarrow{a.s.} T_{CVM}(F_1, F_2),$$

where $T_{CVM}(F_1, F_2) = \int_{-\infty}^{\infty} [F_1(x) - F_2(x)]^2 d(\tau F_1(x) + (1-\tau)F_2(x))$ *is the population version of* $T_{CVM}(\widetilde{F}_1, \widetilde{F}_2)$. *Here,* r_1 *and* r_2 *are the detected edge pixels in the given two images.*

Proposition 5.3.4 justifies the consistency of the proposed test statistic based on the radial distances of the image object in two images. Proposition 5.3.4 can be proved easily using Corollary 5.3.3. A sketch of the proof of Proposition 5.3.4 is provided in Appendix A.4.

5.4 Numerical Studies

In this section, we numerically investigate the performance of the proposed method in comparison with two state-of-the-art methods. We conduct comparative analyses using various simulated toy images. In literature, most of the intensity and feature based image monitoring approaches are overly sensitive to even a little variation in intensities within a region. Consequently, these techniques have a very high probability of false detection since they are highly sensitive to noise. In the manufacturing industry, any small change in intensity values in a region is very common and often correctly interpreted as faults.

However, in many applications such as satellite imaging and medical imaging, such small changes should not be considered as meaningful changes in the underlying image process. This makes the proposed method especially helpful for tracking the shape and size of the image object in satellite and medical imaging. Section 5.2 provides discussions on image comparison and image monitoring individually. In this section also, separate comparative studies with the state-of-the art methods are described for both image comparison and image monitoring. To assess the performance of the competing procedures, we consider empirical size and power of these procedures. In addition, we consider the mean and standard deviation of the out-of-control run length as the performance evaluation measures for online monitoring. Moreover, as the control chart (5.2.7) at ℓ-th time point depends on the images of previous time points, steady-state average run length ARL_1 is the natural choice in the online monitoring stage.

5.4.1 Comparison with state-of-the-art methods

To investigate the numerical performance of the proposed image comparison algorithm, we consider Wang and Ye (2010) as a competing method. The central idea behind the competing method is based on nonparametric comparison of regression surface. It considers the $\mathcal{L}^2$-distance between the surface estimate at each pixel coordinates. The test statistic is thus defined as

$$T_N = \frac{1}{n^2} \sum_{i,j=1}^{n} (\widehat{f}_1(x_i, y_j) - \widehat{f}_2(x_i, y_j))^2, \tag{5.4.1}$$

where $\widehat{f}_1(x_i, y_j)$ and $\widehat{f}_2(x_i, y_j)$ are the local estimators of the two image intensity functions at the pixel coordinate (x_i, y_j). The above test statistic is a multi-dimensional generalisation of the test statistic defined in Dette and Neumeyer (2001). Additionally, for numerical comparison for the online monitoring part, we choose the approach proposed by Zang and Qiu (2018b) as our competitor. The competing method needs additional rigid-body image registration before the monitoring step, i.e., after getting each image in Phase II, we have to estimate one rotation parameter and two translation parameters, which make this method computationally expensive. However, the proposed method does not need any such step, and thus the flexibility and applicability of our method is well justified. The competing methods are comparable in the sense that they do not take into account the correlation structure of the images, they assume independence of the images captured at different time points. Suppose in Phase II stage, after rigid-body image registration, the geometrically aligned surfaces are denoted as $\{\widehat{f}^*_\ell(x, y) : \ell = 1, 2, \ldots\}$. The overall

difference between ℓ-th aligned surface and the in-control image is expressed as

$$\Lambda_{ART,\ell} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}|\widehat{f}_{\ell}^{*}(x_i, y_j) - \widehat{f}_0(x_i, y_j)|, \tag{5.4.2}$$

where $\widehat{f}_0$ can be estimated by combining all jump preserving estimated in-control image surfaces after registering them properly. Then, the control chart statistic based on $\Lambda_{ART,\ell}$ can be expressed in a similar fashion as in (5.2.7). For more discussion regarding this competing method, the readers are referred to Zang and Qiu (2018a,b). Since all competing methods require an initial image registration step, we utilize the R package *RNiftyReg* [Clayden et al. (2023)] to geometrically align the observed images.

5.4.2 Simulation studies

In this part, we present some simulation results regarding the numerical performance of the proposed method of image comparison and image monitoring. Throughout this section, if there is no further specification, Gaussian noise was added to generate observed images. In our simulation study, the true in-control (IC) shape of the image object is assumed to be an ellipse (see Figure 5.5(a)) and the image intensity function of true IC image can be expressed as

$$f_0(x, y) = \mathbb{I}\left(\sqrt{\left(\frac{x-0.5}{0.1875}\right)^2 + \left(\frac{y-0.5}{0.25}\right)^2} < 1\right), \tag{5.4.3}$$

where $(x, y) \in [0, 1] \times [0, 1]$ and $\mathbb{I}(.)$ is the indicator function. For detecting the edge pixels we use the algorithm described in Section 5.2.1. As observed image intensities contain noise, there is a possibility of getting false jump points in the background of the image object and they affect the performance of the proposed method. To get rid of this problem, we use the following algorithm which is similar to the one suggested by Roy and Mukherjee (2024a).

Algorithm for removing false edges

$\mathbf{E}^{n\times n}$ = Binary Edge Matrix
$\mathbf{E_p}^{(n+2p)\times(n+2p)}$ = Padded Edge Matrix (Pad Length= p)
for $i := (1+p)$ to $(n+p-1)$ **do**
  for $j := (1+p)$ to $(n+p-1)$ **do**
    if $\mathbf{E_p}[i,j] \leftarrow 1$ & $\sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1}\mathbf{E_p}[k,l] \leq 4$ **then**
      $\mathbf{E_p}[i,j] \leftarrow 0$
    end if
  end for
end for

We evaluate the performance of the image comparison methods by empirical size and power. We use the following image intensity functions to conduct the simulation study.

1. Only rigid body transformation (i.e., rotation & translation):

$$g_1^{(IC)}(x,y) = \mathbb{I}\left(\sqrt{\left(\frac{x-60/128}{32/128}\right)^2 + \left(\frac{y-60/128}{24/128}\right)^2} < 1\right).$$

2. Smooth change in the background of the image object:

$$g_2^{(IC)}(x,y) = \begin{cases} 1 & \text{if } \sqrt{\left(\frac{x-60/128}{32/128}\right)^2 + \left(\frac{y-60/128}{24/128}\right)^2} < 1 \\ \frac{1}{10}\sin(\frac{x+y}{10}) & \text{otherwise.} \end{cases}$$

3. Change in size of the image object:

$$g_3^{(OC)}(x,y) = \mathbb{I}\left(\sqrt{\left(\frac{x-60/128}{32.5/128}\right)^2 + \left(\frac{y-60/128}{24/128}\right)^2} < 1\right)$$

4. In addition to the rigid transformation change in the size of the image object:

$$g_4^{(OC)}(x,y) = \mathbb{I}\left(\sqrt{\left(\frac{x-60/128}{24.5/128}\right)^2 + \left(\frac{y-60/128}{32/128}\right)^2} < 1\right)$$

5. Change in the shape of the image object:

$$g_5^{(OC)}(x,y) = \mathbb{I}\left(\left\{\sqrt{\left(\frac{x-60/128}{32/128}\right)^2 + \left(\frac{y-60/128}{24/128}\right)^2} < 1\right\} \bigcap \left\{x+y > 154/128\right\}\right)$$

Note that the image intensity functions $g_1^{(IC)}(x,y)$ and $g_2^{(IC)}(x,y)$ imply situations under the null hypothesis, i.e., the shape and size of the image object is unchanged. The intensity functions $g_3^{(IC)}(x,y)$, $g_4^{(IC)}(x,y)$, and $g_5^{(IC)}(x,y)$ correspond to the alternative hypothesis. In all cases, (x,y) are generated from a regular grid of values in the design space $\Omega = [0,1] \times [0,1]$ with a fixed and equally spaced design. Figures 5.5 and 5.6 show various types of null and alternative images as in the situations 1-5 described above. We perform the simulation study under the following settings: (i) two different image resolutions: 128×128 and 256×256, and (ii) two different noise levels of σ: 0.02 and 0.05 for each image resolution. Tables (5.1) illustrates the empirical size and power across various bandwidth parameter values. As outlined in Section 5.2.2, we determine the procedure

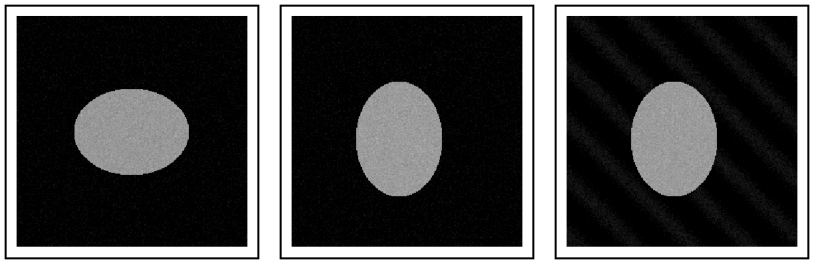


Figure 5.5: In-control simulated images: (a) noisy reference image, (b) the same image object after rigid-body transformation, and (c) the same image object with a smooth change in the background.

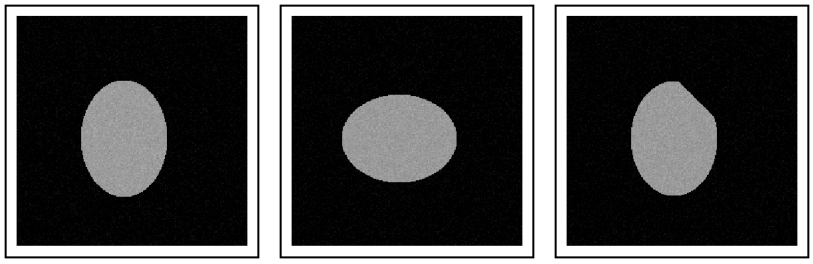


Figure 5.6: Out-of-control simulated images: (a) change in size of the image object, (b) change in size of the image object in presence of rotation and translation, and (c) change in shape of the image object.

parameter h_n using cross-validation procedure. Figure 5.7 presents the cross-validation (CV) scores versus nh_n for the simulated images. From Figure 5.7 it is clear that the optimal bandwidth should be $\widetilde{h}_n = \frac{2}{n}$. Table 5.2 presents the comparative analysis with Wang and Ye (2010). Regarding the competing method, we can see that it always rejects the null hypothesis even when it is true. That is because of the erroneous image registration performance beforehand. Therefore, these type of method that requires image registration step beforehand should be avoided in practice. However, the proposed algorithm exhibits robust performance and consistently outperforms the alternative method by Wang-Ye in all the scenarios mentioned above.

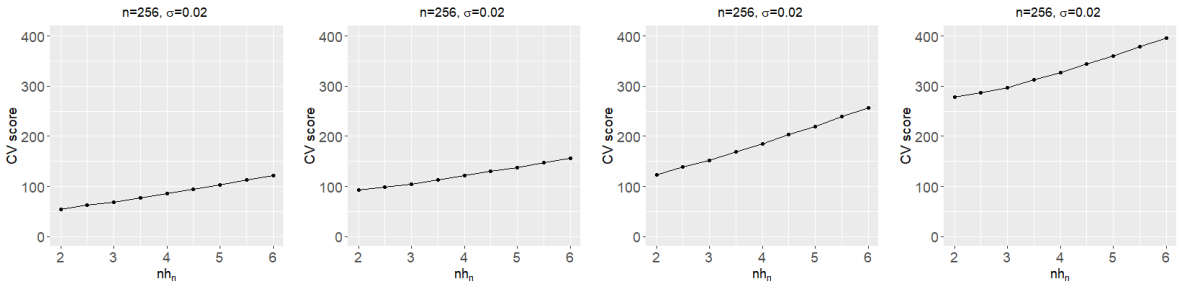


Figure 5.7: Parameter selection based on cross-validation.

Subsequently, we conduct a series of simulations to evaluate the effectiveness of the proposed approach in the context of online monitoring of the shape and size of the image

	$\sigma^2 = 0.02$			$\sigma^2 = 0.05$		
$n = 128$	$nh_n = 2$	$nh_n = 3$	$nh_n = 4$	$nh_n = 2$	$nh_n = 3$	$nh_n = 4$
$g_1^{(IC)}(x, y)$	0.040	0.041	0.047	0.049	0.043	0.045
$g_2^{(IC)}(x, y)$	0.042	0.041	0.151	0.044	0.041	0.093
$g_3^{(OC)}(x, y)$	1.000	1.000	1.000	1.000	1.000	1.000
$g_4^{(OC)}(x, y)$	1.000	1.000	1.000	1.000	1.000	1.000
$g_5^{(OC)}(x, y)$	0.771	1.000	1.000	0.510	0.999	1.000
	$\sigma^2 = 0.02$			$\sigma^2 = 0.05$		
$n = 256$	$nh_n = 2$	$nh_n = 3$	$nh_n = 4$	$nh_n = 2$	$nh_n = 3$	$nh_n = 4$
$g_1^{(IC)}(x, y)$	0.058	0.061	0.046	0.061	0.061	0.064
$g_2^{(IC)}(x, y)$	0.059	0.035	0.186	0.057	0.010	0.080
$g_3^{(OC)}(x, y)$	1.000	1.000	1.000	1.000	1.000	1.000
$g_4^{(OC)}(x, y)$	1.000	1.000	1.000	1.000	1.000	1.000
$g_5^{(OC)}(x, y)$	0.999	1.000	1.000	1.000	1.000	1.000

Table 5.1: Empirical sizes and powers of the proposed method for various images using different values of the bandwidth parameter h_n.

object. For this purpose, we consider the test image in figure 5.6 with resolution 128×128 (i.e., $n = 128$) and generate $m = 36$ Phase I images by rotating the image object by equally spaced rotation angles. To each image, we add a zero-mean Gaussian noise with standard deviation (σ) equaling 0.02. In this study, we fix the nominal in-control $ARL_0 = 20$. Note that the number of Phase I images and the pre-defined ARL_0 value in our experiment are relatively small compared to the conventional SPC literature. This is primarily due to limited availability of similar images in practical scenarios. To set the control limit at a pre-fixed ARL_0, we use the bootstrap technique described in Section 5.2.3. We divide the simulation performance in two parts. The first part evaluates the in-control performance, and the second one evaluates the out-of-control performance. Based on the definition of *average run length*, we anticipate a high value of ARL_1, i.e., close to the value of pre-fixed $ARL_0 = 20$ for the in-control images, and a substantially low value of ARL_1 for the out-of-control images.

In-control performance: Based on in-control image samples, we numerically compare the performance of the proposed method with the state-of-the-art method proposed by Zang and Qiu (2018b) which we call Zang-Qiu hereafter. When we refer to in-control image samples, we mean that the shape and size of the image object remain unchanged but there could be rigid-body transformation such as rotation and/or translation of the image object. See Figure 5.5 for various in-control images. Another intriguing situation with potential application is depicted in Figure 5.5(c) where there is a smooth change in

$(n = 128, h_n = \frac{3}{n})$	$\sigma^2 = 0.02$		$\sigma^2 = 0.05$	
	Proposed	Wang-Ye	Proposed	Wang-Ye
$g_1^{(IC)}(x, y)$	0.041	1.000	0.043	1.000
$g_2^{(IC)}(x, y)$	0.041	1.000	0.041	1.000
$g_3^{(OC)}(x, y)$	1.000	1.000	1.000	1.000
$g_4^{(OC)}(x, y)$	1.000	1.000	1.000	1.000
$g_5^{(OC)}(x, y)$	1.000	1.000	0.999	1.000
$(n = 256, h_n = \frac{3}{n})$	$\sigma^2 = 0.02$		$\sigma^2 = 0.05$	
	Proposed	Wang-Ye	Proposed	Wang-Ye
$g_1^{(IC)}(x, y)$	0.061	1.000	0.061	1.000
$g_2^{(IC)}(x, y)$	0.035	1.000	0.010	1.000
$g_3^{(OC)}(x, y)$	1.000	1.000	1.000	1.000
$g_4^{(OC)}(x, y)$	1.000	1.000	1.000	1.000
$g_5^{(OC)}(x, y)$	0.999	1.000	1.000	1.000

Table 5.2: Empirical sizes and powers of the proposed method and its competitor for various in-control and out-of-control images.

the background of the image object. The first two rows of Table 5.3 show the average run length for both, the proposed method and the competing method, based on 1000 replications. We perform out the simulation experiment using two values of the allowance parameter κ. As the images are in-control, it is expected that the value of the ARL_1 should be close to ARL_0. Across all the scenarios of the this example, our proposed algorithm emerges as the superior choice. The competing method performs comparatively fairly only in the case of rigid-body transformation. In presence of smooth changes in the background, competing method fails to disregard it appropriately. Therefore, we should be more cautious to use the competing method in real-life scenarios. In contrast, the proposed method consistently achieves ARL_1 value close to the prefixed ARL_0 value in the situations mentioned in this example for in-control images. It ignores unimportant alternation of foreground and background image intensity values quite well. Hence, according to the in-control performance, the proposed method outperforms its competitor in each of the aforementioned scenarios.

Out-of-control performance: Now, we evaluate the out-of-control performance of the proposed method in comparison with Zang-Qiu. Figure 5.6 shows various types of out-of-control test images. Out-of-control images contain alterations of the shape and/or size of the image object even after accommodating rigid-body transformations, if there is any. Table (5.3) shows ARL_1 values based on 1000 replications for two values of the allowance parameter κ. In each replication of run length, we choose the first 4 images to be in-

$(n = 128, h_n = \frac{3}{n})$	Proposed $\kappa = 0.1$	Proposed $\kappa = 0.5$	Zang-Qiu $\kappa = 0.1$	Zang-Qiu $\kappa = 0.5$
$g_1^{(IC)}(x, y)$	15.76 (5.46)	14.84 (6.57)	12.03(1.02)	11.09 (1.05)
$g_2^{(IC)}(x, y)$	18.38 (7.40)	19.31 (9.95)	7.95 (0.47)	7.33 (0.51)
$g_3^{(OC)}(x, y)$	1.09 (0.29)	1.00 (0.00)	7.70 (1.01)	6.76 (1.03)
$g_4^{(OC)}(x, y)$	1.01 (0.10)	1.00 (0.00)	8.19 (0.98)	7.22 (1.03)
$g_5^{(OC)}(x, y)$	1.00 (0.00)	1.00 (0.00)	7.29 (0.87)	6.35 (0.91)

Table 5.3: Mean ARL of the proposed method and its competitor for various simulated in-control and out-of-control images.

control, and subsequent images to be out-of-control. If a method raises signal before the 5-th time point, we exclude the replication from ARL_1 calculation. From Table (5.3) we see that the proposed method has a significantly better performance in each case of this example as compared to Zang-Qiu. The proposed algorithm raise a signal at the earliest possible time, while the competing method has higher values of ARL_1 in each case of the current example. Note that, a larger ARL_1 in the presence of a sequence of out-of-control images implies more detection delay.

5.5 Real Data: Monitoring Salton Sea Area

In this section, we present the performance of the proposed method for the surveillance real images. We collect the image data of the Salton Sea area from the LANDSAT project (Data source: https://earthexplorer.usgs.gov/). As discussed in Section 5.1, the area of Salton sea is shrinking gradually over time and this is due to the population growth, water demand or weather condition. The local ecosystem of animals and plants

Figure 5.8: Cropped satellite images of the Salton sea area captured in 2014 (left), 2018 (middle), and 2022 (right).

[Shuford et al. (2002)] are badly impacted by the shrinkage of Salton sea. It is the largest inland lake at the southernmost part of California. In this example, our primary objective is to monitor the shape and size of Salton sea using radial distances. Figure 5.8 shows

the Salton sea area in 2014, 2018, and 2022. For image surveillance, we consider the images of Salton Sea captured at the second quarter of each year from 2014 to 2023. We consider the images captured in 2014 as in-control images and construct the control chart as discussed in Section 5.2.3. We generate 30 Phase I sample images by employing rigid-body transformations on the in-control images. For testing purpose, we use the images captured at the second quarter from 2015 to 2023. In this study, we fix the in-control ARL to be 20. Now, we are at the stage of monitoring the sequence of images of the Salton sea region. Firstly, we detect the boundary curve by the LLK method discussed in Section 5.2.1, and estimate the location of the centroid by the sample mean of the coordinates of the detected edge pixels. As the proposed method does not directly depend on the image intensity values, it can ignore the small intensity changes due to the presence of clouds in the satellite images of Salton sea. Moreover, we ignore the falsely detected edge pixels by the algorithm mentioned in Section 5.4.2. In Phase II, we find that the mean of the steady state ARL_1 is equals to 3.52 with the standard deviation of the run lengths equaling 0.687, while the corresponding median of ARL_1 being 4, based on 500 replications. Therefore, the first signal given by the proposed CUSUM method is at the second quarter of 2018. Figure 5.9 shows the result of online monitoring by the proposed method.

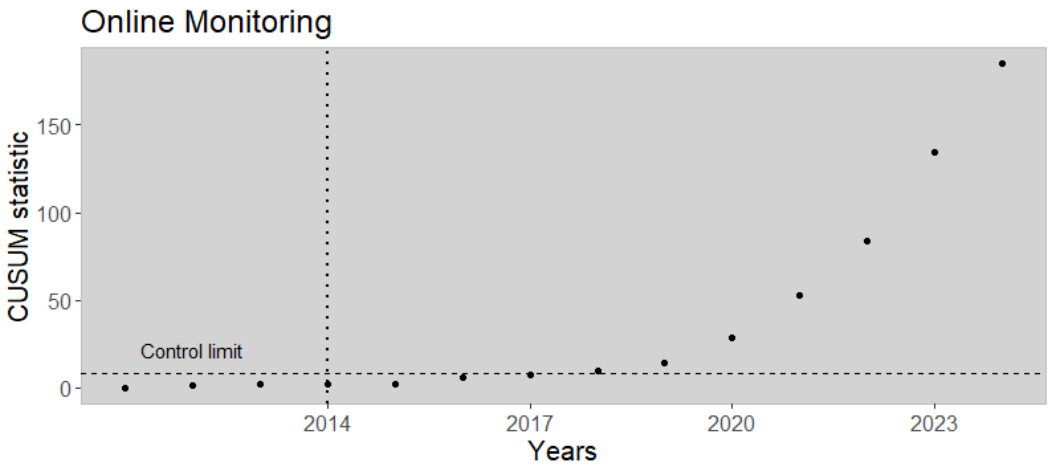


Figure 5.9: Online monitoring of Salton sea images. The horizontal line indicates the control limit and the image samples before the vertical line are assumed as in-control.

Note that, we are unable to find any other study based on the recent LANDSAT dataset of Salton sea area from 2014-2023. In literature [Jones and Fleck (2020); Yi and Qiu (2023)], there exists a number of studies that validate the long-term decline of Salton Sea shoreline. Nonetheless, the proposed algorithm shows a notable shift in the time period from 2014 to 2018. See Figure 5.10 for year-wise plot of the distribution of the radial distance. We believe that further ecological research is necessary to investigate such a rapid decline of the Salton sea area.

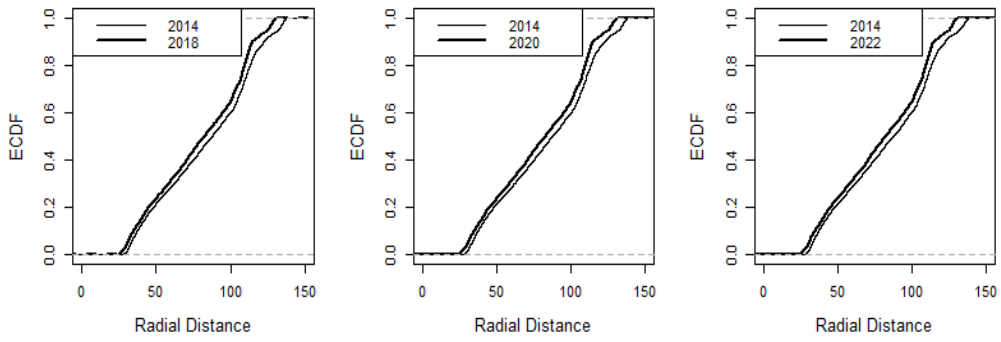


Figure 5.10: Estimated cumulative distribution functions (ECDFs) of the radial distances of the Salton sea images.

5.6 Concluding Remarks

Motivated by the issue of image registration while comparing or monitoring images such as satellite images, we develop a novel procedure to compare and monitor the shape and size of an image object in presence of rigid-body transformation. In literature, this is a well-known and well-discussed problem. However, as per our knowledge, this is one of the first attempts to bypass the step of image registration, and address the problem of image comparison and monitoring appropriately. Numerical studies and theoretical justifications show that it has an effective and reliable performance in various scenarios examined.

Below we point out a number of issues that we have not taken into account in this chapter regarding the proposed method, and are worthy of further investigations. To begin with, the test based on equality of the distributions of the radial distances is not equivalent to the test of equality of the shapes and sizes of an image object in two images. It is theoretically possible to intelligently construct image objects of different shapes but having the same distribution of the radial distances. However, such examples are rare in practical situations. In reality, the proposed method is useful and capable to detect changes of the shape and/or size of a large class of image objects. Secondly, only rigid-body transformation is assumed in this chapter. In satellite imaging, scaling or zooming transformation [Das et al. (2024)] is also very widespread. Therefore, scale invariant extension of the proposed method will be useful in practice. Lastly, the proposed method assumes that the images over time are independent. However, in practice, there could be serial correlation or other data dependence among the images. Extensive research is needed to address these issues.

Chapter 6

Concluding Remarks and Future Directions

In this dissertation, we introduce innovative image comparison and monitoring schemes to tackle various issues that have not been addressed in the literature yet. The core strength of this dissertation lies in its simplicity, novel ideas, and wide-ranging applications to real-world data scenarios. Remembering the comment by renowned statistician, Prof. Prasanta Chandra Mahalanobis, "Statistics is the universal tool of inductive inference, research in natural and social sciences, and technological applications," we believe that our methodological advancements will be highly beneficial to practitioners in various fields.

6.1 Limitations And Future Scopes

In this section, we discuss some limitations of our work presented in this dissertation. The current contributions primarily focus on the rapid detection of faults and anomalies through lower ARLs rather than estimating the sizes and locations of faults within the image. However, estimating the fault size and location are important aspects in the field of image monitoring/surveillance. An important avenue for future work would be to extend these methods to estimate the locations and sizes of the faults. Throughout the dissertation, we have assumed that the time-varying images are independent; however, it would be more practical to accommodate spatio-temporal correlation in the observed image sequence. In literature, there are several studies which attempt to account for spatial correlation in the images, for example Otto (2019) and Okhrin et al. (2020). However, the discussion remains limited when it comes to jump-preserving surveillance of spatio-temporal image sequences. Recently, Yi and Qiu (2021) provide jump preserving image denoising procedure in the presence of spatio-temporal correlation structure. Extending

our current research to monitor spatio-temporal image sequences would be a valuable future direction. In most of the chapters, we have assumed that images are aligned geometrically. In chapter 5, we propose an algorithm for monitoring the shape and size of an image object that can handle image misalignment issues by its construction. Nonetheless, significant future research is needed to tackle this issue for more general images. Furthermore, in some image monitoring applications, the in-control distribution of image sequences might change over time due to seasonality. This type of situation is common in satellite imaging and mostly opens areas for further investigation. Image monitoring scheme, accommodating time-varying in-control distribution needs much future research effort.

6.2 Possible Future Research Areas Related to Monitoring

Research on monitoring network data, surveillance of Riemannian functional data, etc., are limited in the literature and currently are important to explore.

Network data monitoring: Network sequence has been frequently used for describing the longitudinal pattern of a dynamic system. In practice, to characterize the state of a network at a given time point, people use tools from graph theory. Since the relationship among nodes in a graph often evolves over time, monitoring network sequence can be viewed as *dynamic* process monitoring. See Stevens et al. (2021a,b); Dong et al. (2020) for more discussions on monitoring network data.

Statistical process monitoring of Riemannian function data: This challenging task involves monitoring functional data where each function value lies on a Riemannian manifold, as opposed to traditional functional data analysis where the observations are in Euclidean space [Dai and Müller (2018); Su et al. (2014)]. This has wide applications in statistical shape monitoring.

Appendix A

Appendix with Additional Details

A.1 Sketches of Proofs of the Stated Propositions in Chapter 2

A.1.1 Proof of Proposition 2.3.1

Using Theorem 1 of Mei (2010), as $\Lambda_0 \to \infty$,

$$E_\infty\Big\{T_{\text{sum}}(\Lambda_0)\Big\} \geq \Big(1+o(1)\Big)\left[\frac{\exp(\Lambda_0)}{1+\Lambda_0+\Lambda_0^2/2!+\ldots+\Lambda_0^{(n^2-1)}/(n^2-1)!}\right].$$

We know that, $T_{\text{sum}}(\Lambda_0) \leq T^{(r)}_{\text{sum}}(\Lambda_0)$. Hence,

$$E_\infty\Big\{T^{(r)}_{\text{sum}}(\Lambda_0)\Big\} \geq E_\infty\Big\{T_{\text{sum}}(\Lambda_0)\Big\},$$

$$\text{i.e., } E_\infty\{T^{(r)}_{\text{sum}}(\Lambda_0)\} \geq \Big(1+o(1)\Big)\left[\frac{\exp(\Lambda_0)}{1+\Lambda_0+\Lambda_0^2/2!+\ldots+\Lambda_0^{(n^2-1)}/(n^2-1)!}\right].$$

■

A.1.2 Proof of Proposition 2.3.2

Define $T_{\text{sum}}(a) = \inf\Big\{t \geq 1 : \sum_{\ell=1}^{n} C_k^{(\ell)} > a\Big\}$ as the stopping time of the SUM-scheme of the CUSUM statistic. Then, by Theorem 1 of Mei (2010), as $a \to \infty$,

$$\overline{E}^{(p_1,p_2,\ldots,p_s)}(T_{\text{sum}}(a)) \leq \left[\frac{2a\sigma^2}{\sum_{(x,y)\in\mathcal{S}}(f_0(x,y)-f_1(x,y))^2}\right] + O(1).$$

Now, we know that $T_{\text{sum}}^{(r)}\left(\frac{a(n^2-r+1)}{n^2}\right) \leq T_{\text{sum}}(a)$. Therefore,

$$\overline{E}^{(p_1,p_2,\ldots,p_s)}\left[T_{\text{sum}}^{(r)}\left(\frac{a(n^2-r+1)}{n^2}\right)\right] \leq \overline{E}^{(p_1,p_2,\ldots,p_s)}(T_{\text{sum}}(a)),$$

$$\text{i.e., } \overline{E}^{(p_1,p_2,\ldots,p_s)}\left[T_{\text{sum}}^{(r)}\left(\frac{a(n^2-r+1)}{n^2}\right)\right] \leq \frac{2a\sigma^2}{\sum_{(x,y)\in\mathcal{S}}(f_0(x,y)-f_1(x,y))^2}.$$

For fixed r and n, as $a \to \infty$, $\Lambda_0 = \left(\frac{a(n^2-r+1)}{n^2}\right) \to \infty$.

Therefore, as $\Lambda_0 \to \infty$, $\overline{E}^{(p_1,p_2,\ldots,p_s)}(T_{\text{sum}}^{(r)}(\Lambda_0)) \leq \frac{2\Lambda_0\left(\frac{n^2}{n^2-r+1}\right)\sigma^2}{\sum_{(x,y)\in\mathcal{S}}(f_0(x,y)-f_1(x,y))^2}$. ■

A.2 Sketches of Proofs of the Stated Propositions in Chapter 3

Lemma A.2.1. *Under the conditions stated in Proposition 3.6.1, we have, for* $i_1, i_2 = 0, 1, 2,$

$$\left\| \frac{1}{n^2h_n^2} \sum \left(\frac{x_i - x}{h_n}\right)^{i_1} \left(\frac{y_j - y}{h_n}\right)^{i_2} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) - \nu_{i_1 i_2} \right\|_{\Omega_{h_n}} = O\left(\frac{1}{nh_n}\right)$$

and

$$\left\| \frac{1}{n^2h_n^2} \sum \varepsilon_{ij} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \right\|_{\Omega_{h_n}} = o\left(\frac{\beta_n \log(n)}{nh}\right) \quad a.s.$$

Proof of Lemma A.2.1: See Proposition 2 of Qiu (2009) for the proof of Lemma A.2.1. ■

Lemma A.2.2. *Under the conditions in Proposition 3.6.1, we have*

$$\|\widehat{a} - f\|_{\Omega_{\bar{E},h_n}} = O\left(h_n^2\right) + o\left(\frac{\beta_n \log n}{nh_n}\right) a.s.$$

$$\left\|\widehat{b} - f_x'\right\|_{\Omega_{\bar{E},h_n}} = O\left(h_n\right) + o\left(\frac{\beta_n \log n}{nh_n^2}\right) a.s.$$

$$\left\|\widehat{c} - f_y'\right\|_{\Omega_{\bar{E},h_n}} = O\left(h_n\right) + o\left(\frac{\beta_n \log n}{nh_n^2}\right) \quad a.s.$$

On the other hand, if $(x, y) \in E_{h_n} \backslash S_\epsilon$, *then we have*

$$\begin{pmatrix} \widehat{a} \\ \widehat{b} \\ \widehat{c} \end{pmatrix} = \begin{pmatrix} f_-(x,y) \\ f'_x(\tilde{x},\tilde{y}) \\ f'_y(\tilde{x},\tilde{y}) \end{pmatrix} + \begin{pmatrix} \phi_0(x,y)C(x,y) + O\left(h_n^2\right) + o\left(\frac{\beta_n \log n}{nh_n}\right) \\ \phi_1(x,y)C(x,y) + \gamma_1(x,y)C_x(x,y) + O\left(h_n\right) + o\left(\frac{\beta_n \log n}{nh_n^2}\right) \\ \phi_2(x,y)C(x,y) + \gamma_2(x,y)C_y(x,y) + O\left(h_n\right) + o\left(\frac{\beta_n \log n}{nh_n^2}\right) \end{pmatrix},$$

where $f_-(x,y)$ *is the smaller of the two one-sided (due to JLC) limits of* f *at* (x,y), $(\tilde{x},\tilde{y})$ *is some point around* (x,y) *that satisfies (i) it is a continuity point of* f *that is on the same side of the JLC as* (x,y), *and (ii)* $D((\tilde{x},\tilde{y}),(x,y)) \sim O(1/n), C(x,y)$, $C_x(x,y), C_y(x,y)$ *are absolute jump magnitudes of* $f(x,y)$ *and its first order* x *and* y *partial derivatives,* $\phi_1(x,y)$ *and* $\phi_2(x,y)$ *are two constants satisfying*

$$\sqrt{\phi_1^2(x,y) + \phi_2^2(x,y)} = O\left(1/h_n\right) \ \ a.s.,$$

$\gamma_1(x,y)$ *and* $\gamma_2(x,y)$ *are two constants between* -1 *and* 1 , *and* $\phi_0(x,y)$ *is a constant between* 0 *and* 1.

Proof of Lemma A.2.2: When $(x,y) \in \Omega_{\bar{E},h_n}$, by Taylor's series expansion, for any $(x_i,y_j) \in B_{h_n}(x,y)$, we have

$$\begin{aligned} w_{ij} &= f\left(x_i,y_j\right) + \varepsilon_{ij} \\ &= f(x,y) + \left(x_i - x\right) f'_x(x,y) + \left(y_j - y\right) f'_y(x,y) + O\left(h_n^2\right) + \varepsilon_{ij}. \end{aligned}$$

So, we have

$$\begin{aligned} \begin{pmatrix} \sum w_{ij}K_{ij} \\ \sum w_{ij}\left(x_i - x\right)K_{ij} \\ \sum w_{ij}\left(y_j - y\right)K_{ij} \end{pmatrix} &= \begin{pmatrix} m_{00} & m_{10} & m_{01} \\ m_{10} & m_{20} & m_{11} \\ m_{01} & m_{11} & m_{02} \end{pmatrix} \begin{pmatrix} f(x,y) \\ f'_x(x,y) \\ f'_y(x,y) \end{pmatrix} \\ &+ \begin{pmatrix} \sum O\left(h_n^2\right)K_{ij} + \sum \varepsilon_{ij}K_{ij} \\ \sum O\left(h_n^2\right)\left(x_i - x\right)K_{ij} + \sum \varepsilon_{ij}(x_i - x)K_{ij} \\ \sum O\left(h_n^2\right)\left(y_j - y\right)K_{ij} + \sum \varepsilon_{ij}(y_j - y)K_{ij} \end{pmatrix}. \end{aligned} \tag{A.2.1}$$

By (3.2.3), (A.2.1) and Lemma A.2.1, we have

$$\begin{pmatrix} \widehat{a} \\ \widehat{b} \\ \widehat{c} \end{pmatrix} = \begin{pmatrix} f(x,y) \\ f'_x(x,y) \\ f'_y(x,y) \end{pmatrix} + \begin{pmatrix} O\left(h_n^2\right) + o\left(\frac{\beta_n \log n}{nh_n}\right) \\ O\left(h_n\right) + o\left(\frac{\beta_n \log n}{nh_n^2}\right) \\ O\left(h_n\right) + o\left(\frac{\beta_n \log n}{nh_n^2}\right) \end{pmatrix} a.s. \tag{A.2.2}$$

The above expression is an application of Lemma A.2.1 and under the condition of Proposition 3.6.1, it is evident that (A.2.2) is uniformly true for $(x,y) \in \Omega_{\bar{E},h_n}$.

Now, consider the case when (x,y) is on JLC and is not a singular point (i.e.,$(x,y) \in$

$E_{h_n} \backslash S_\epsilon$). When n is large enough so that $h_n < \epsilon$, then $B_{h_n}(x, y)$ is divided into two regions by JLC defined as I_1 and I_2. Without loss of generality we can assume that there is a positive jump at (x, y) from I_1 to I_2. For $(x_i, y_j) \in I_1$, we have

$$\begin{aligned} w_{ij} &= f(x_i, y_j) + \varepsilon_{ij} \\ &= f_-(x, y) + (x_i - x) f'_x(\tilde{x}, \tilde{y}) + (y_j - y) f'_y(\tilde{x}, \tilde{y}) + O\left(h_n^2\right) + \varepsilon_{ij}. \end{aligned}$$

Similarly, when $(x_i, y_j) \in I_2$, we have

$$\begin{aligned} w_{ij} &= f(x_i, y_j) + \varepsilon_{ij} \\ &= f_-(x, y) + (x_i - x) f'_x(\tilde{x}, \tilde{y}) + (y_j - y) f'_y(\tilde{x}, \tilde{y}) \\ &+ C(x, y) + (x_i - x) C_x(x, y) + (y_j - y) C_y(x, y) \\ &+ O\left(h_n^2\right) + \varepsilon_{ij}. \end{aligned}$$

Here $(\tilde{x}, \tilde{y})$ is some point around (x, y) that satisfies the condition mentioned in Lemma A.2.2. Now, by (3.2.3) and above expression using similar argument of (A.2.2), we have

$$\begin{pmatrix} \widehat{a} \\ \widehat{b} \\ \widehat{c} \end{pmatrix} = \begin{pmatrix} f_-(x, y) \\ f'_x(\tilde{x}, \tilde{y}) \\ f'_y(\tilde{x}, \tilde{y}) \end{pmatrix} + \begin{pmatrix} \phi_0(x, y) C(x, y) + O\left(h_n^2\right) + o\left(\frac{\beta_n \log n}{n h_n}\right) \\ \phi_1(x, y) C(x, y) + \gamma_1(x, y) C_x(x, y) + O(h_n) + o\left(\frac{\beta_n \log n}{n h_n^2}\right) \\ \phi_2(x, y) C(x, y) + \gamma_2(x, y) C_y(x, y) + O(h_n) + o\left(\frac{\beta_n \log n}{n h_n^2}\right) \end{pmatrix},$$

where

$$\begin{aligned} \phi_0(x, y) &= \frac{\sum_{(x_i, y_j) \in I_2} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}{\sum K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}, \\ \phi_1(x, y) &= \frac{\sum_{(x_i, y_j) \in I_2} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}{\sum (x_i - x)^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}, \\ \gamma_1(x, y) &= \frac{\sum_{(x_i, y_j) \in I_2} (x_i - x)^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}{\sum (x_i - x)^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}, \\ \phi_2(x, y) &= \frac{\sum_{(x_i, y_j) \in I_2} (y_j - y) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}{\sum (y_j - y)^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}, \\ \gamma_2(x, y) &= \frac{\sum_{(x_i, y_j) \in I_2} (y_j - y)^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}{\sum (y_j - y)^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)}. \end{aligned}$$

From the above expression, it is clear that γ_1 and γ_2 are always positive constants between 0 and 1 and ϕ_0 is a constant between 0 to 1. From Lemma A.2.2, we know that $C_x(x, y)$ and $C_y(x, y)$ is the absolute jump magnitude of f'_x and f'_y respectively, therefore

γ_1 and γ_2 are constants between -1 and 1. By similar idea of Qiu and Yandell (1997) it is easy to check $\sqrt{\phi_1^2(x,y)+\phi_2^2(x,y)} = O\left(1/h_n\right)$ a.s.. Hence Lemma A.2.2 is proved. ■

A.2.1 Proof of Proposition 3.6.1

For a design point $(x,y) \in \Omega_{\bar{S},\epsilon}$, if it is more than h_n away from any JLC, then at least one of $B_{h_n}\left(x_{P_1}, y_{P_1}\right)$ and $B_{h_n}\left(x_{P_2}, y_{P_2}\right)$ is located in a same continuous region as (x,y). So, we have

$$\lambda(x,y) \leq \left\| \widehat{\beta}(x,y) - \widehat{\beta}_{N_1}(x,y) \right\| = O\left(h_n\right) + o\left(\frac{\beta_n \log n}{n h_n^2}\right) \text{ a.s.}$$

The above expression is a direct application of Lemma A.2.2. From the fact $\chi^2_{2,(1-\alpha_n)} \leq 2\chi^2_{1,(1-\alpha_n)/2}$ and the Mill's inequality regarding normal tail probabilities it can be shown easily that

$$\chi^2_{2,(1-\alpha_n)} = O\left(-\log(1-\alpha_n)\right).$$

Using the above fact, Lemma A.2.1, and the expression of threshold value, it is not difficult to check that $v_n = O\left(\frac{n\sqrt{-\log(1-\alpha_n)}}{(nh_n^2)^{5/2}}\right)$ a.s. So under the condition $\frac{(nh_n)^{7/2}}{n^2\sqrt{-\log(1-\alpha_n)}} = o(1)$, we have $\frac{\lambda(x,y)}{v_n} = o(1)$ a.s. which implies $\lambda(x,y) < v_n$ a.s. The above expression establishes the fact that when n is large enough, a design point (x,y) on the continuous region is not detected as an edge point, and this is uniformly true for all $(x,y) \in \Omega_{\bar{S},\epsilon} \cap \Omega_{\bar{E},h_n}$. Therefore,

$$\sup_{(x,y)\in\widehat{E}_n\cap\Omega_{\bar{S},\epsilon}} \inf_{(x',y')\in E\cap\Omega_{\bar{S},\epsilon}} D\left((x,y)^T,(x',y')^T\right) = O\left(h_n\right) \text{ a.s.} \tag{A.2.3}$$

Now, consider the case where (x,y) is a non-singular point on a JLC, Then, by Lemma A.2.2, we have

$$\lambda(x,y) \sim C(x,y)\sqrt{\phi_1^2(x,y)+\phi_2^2(x,y)} + O\left(h_n\right) + o\left(\frac{\beta_n \log n}{n h_n^2}\right) \text{ a.s.}$$

From Lemma A.2.2, we know that $\sqrt{\phi_1^2(x,y)+\phi_2^2(x,y)} = O\left(1/h_n\right)$ a.s. . By the condition $\sqrt{\frac{-log(1-\alpha_n)}{(nh_n)^3}} = o(1)$ from Proposition 3.6.1, we have $\lambda(x,y) > v_n$ a.s.. Therefore, (x,y) would be detected almost surely as edge points when n is large enough. By the definition of singular point discussed in Section 3.6, since $\min_{(x,y)\in D\cap\Omega_{\bar{S},\epsilon}} C(x,y) > 0$, the above result is uniformly true for $(x,y) \in E \cap \Omega_{\bar{S},\epsilon}$.
Therefore,

$$\sup_{(x,y)\in E\cap\Omega_{\bar{S},\epsilon}} \inf_{(x',y')\in\widehat{E}_n\cap\Omega_{\bar{S},\epsilon}} D\left((x,y)^T,(x',y')^T\right) = O\left(h_n\right) \text{ a.s.} \tag{A.2.4}$$

By (A.2.3) and (A.2.3), Proposition 3.6.1 is proved. ■

A.3 Sketches of Proofs of the Stated Propositions in Chapter 4

Assume that there exists a finite partition $\{\Gamma_l, l = 1, 2, \dots P\}$ of the design space $[0,1] \times [0,1]$ such that (i) each Γ_l is a connected region in the design space. (ii) Define $\partial\Gamma_l$ as the set of boundary points in Γ_l and $f(x,y)$ is continuous in $\Gamma_l \setminus \partial\Gamma_l$, for $l = 1, 2, \dots, P$. (iii) $\bigcup_{i=1}^{P} \Gamma_l = \Omega$. (iv) There exists at most finitely many points $\{(x_k^*, y_k^*), k = 1, 2, \dots, K^*\}$ in $[\bigcup_{i=1}^{n} \partial\Gamma] \bigcap \Omega$ such that for each (x^*, y^*) with $k = 1, 2, \dots K^*$, there are $\Gamma_{k_1}^*, \Gamma_{k_2}^* \in \{\Gamma_l, l = 1, 2, \dots P\}$ satisfying (a) $(x^*, y^*) \in \Gamma_{k_1}^* \bigcap \Gamma_{k_2}^*$, and (b) $\lim_{(x,y)\to(x^*,y^*),(x,y)\in\Gamma_{k_1}^*} f(x,y) = \lim_{(x,y)\to(x^*,y^*),(x,y)\in\Gamma_{k_2}^*} f(x,y)$. In the JRA literature, we call $[\bigcup_{i=1}^{n} \partial\Gamma] \bigcap \Omega$ the jump location curves (JLCs) of the discontinuous image regression surface.

Lemma A.3.1. *Under the conditions in Proposition 4.4.1, if* $(x,y) \in \Omega_{\bar{E},\epsilon}$, *then*

$$S(x, y, h_n) \stackrel{a.s.}{\to} \sigma.$$

Proof of Lemma A.3.1: As $(x,y) \in \Omega_{\bar{E},\epsilon}$, the circular neighborhood $B(x, y; h_n)$ lies in a continuous region. Then, the pixel intensities in $B(x, y; h_n)$ are *i.i.d.* with $E(w_{xy}^{\ell}) = \mu$ (say), and $var(w_{xy}^{\ell}) = \sigma^2$, where $\{w_{xy}^{\ell} : \ell = 1, 2, \dots, |B(x.y; h_n)|\}$ are the observed pixel intensities in the circular neighborhood $B(x, y; h_n)$. Now,

$$\begin{aligned}
S^2(x, y, h_n) &= \frac{1}{|B(x, y; h_n)|} \sum_{\ell} (w_{xy}^{\ell} - \bar{w}_{xy})^2 \\
&= \frac{1}{|B(x, y; h_n)|} \sum_{\ell} \{(w_{xy}^{\ell} - \mu) + (\mu - \bar{w}_{xy})\}^2 \\
&= \frac{1}{|B(x, y; h_n)|} \sum_{\ell} (w_{xy}^{\ell} - \mu)^2 + (\mu - \bar{w}_{xy})^2 + 2(\mu - \bar{w}_{xy}) \frac{1}{|B(x, y; h_n)|} \sum_{\ell} (w_{xy}^{\ell} - \mu) \\
&= \frac{1}{|B(x, y; h_n)|} \sum_{\ell} (w_{xy}^{\ell} - \mu)^2 - (\mu - \bar{w}_{xy})^2
\end{aligned}$$

By the strong law of large numbers (SLLN), $\frac{1}{|B(x,y;h_n)|} \sum_{\ell} (w_{xy}^{\ell} - \mu)^2 \stackrel{a.s.}{\longrightarrow} \sigma^2$ and $\bar{w}_{xy} \stackrel{a.s.}{\longrightarrow} \mu$. Therefore, by the additive property of almost sure convergence, $S(x, y, h_n) \stackrel{a.s.}{\to} \sigma$. ■

A.3.1 Proof of Proposition 4.4.1:

Note that from Lemma A.3.1, if $(x, y) \in \Omega_{\bar{E},\epsilon}$, then $S(x, y, h_n) \xrightarrow{a.s.} \sigma$. Similarly, $\widehat{\sigma} \xrightarrow{a.s.} \sigma$. Hence the first statement is correct. Now consider the situation where $(x, y) \in E_{h_n}$, i.e., there are two clusters in $B(x, y; h_n)$. Then,

$$S^2(x, y, h_n) = \frac{n_1 s_1^2 + n_2 s_2^2}{n_1 + n_2} + \frac{n_1 n_2}{(n_1 + n_2)^2}(\bar{x}_1 - \bar{x}_2)^2,$$

where $\bar{x}_i$ and s_i^2 represents the sample mean and sample variance of the intensities from the i-th segment in $B(x, y; h_n)$, for $i = 1, 2$. Now, using a similar argument as in the proof of Lemma A.3.1, it can be shown that $\frac{n_1 s_1^2 + n_2 s_2^2}{n_1 + n_2} \xrightarrow{a.s.} \sigma^2$. Note that $(\bar{x}_1 - \bar{x}_2)^2 \xrightarrow{a.s.} (JS)^2$, where JS is the jump size. For $\kappa = 1$, under the conditions in Proposition 4.4.1, the statement is correct under any positive JS. Similarly, for a larger κ, the statement is also correct when JS is large enough. ■

Proof of Corollary 4.4.3: For $(x, y) \in E_{h_n}$, if we wish not to perform local clustering in the circular neighborhood $B(x, y; h_n)$, then using (i) of Proposition 4.4.1,

$$S^2(x, y, h_n) = \frac{n_1 s_1^2 + n_2 s_2^2}{n_1 + n_2} + \frac{n_1 n_2}{(n_1 + n_2)^2}(\bar{x}_1 - \bar{x}_2)^2 \leq \kappa^2 \widehat{\sigma}^2.$$

Now, $(\bar{x}_1 - \bar{x}_2)^2 \xrightarrow{a.s.} \zeta^2$, $S(x, y, h_n) \xrightarrow{a.s.} \sigma$, and $\widehat{\sigma} \xrightarrow{a.s.} \sigma$. Therefore, from the above inequality, it is clear that the proposed algorithm almost surely decides not to cluster if $\kappa \geq \sqrt{(1 + \zeta^2/\sigma^2)}$. ■

To prove Proposition 4.4.4, we start with the following Lemma.

Lemma A.3.2. *Under H_0, for an arbitrary pixel $(x, y) \in B(x, y; h_n)$,*

$$H(\mathcal{C}(x, y)) - I(\mathcal{C}(x, y), \mathcal{C}'(x, y)) = O(\delta) \ \ a.s.$$

Proof of Lemma A.3.2: Here we are assuming that $(x, y) \in E_{h_n}$, i.e., $B(x, y; h_n)$ contains a JLC. The situation when (x, y) is in a continuity region, the proof will be similar and trivial.

Define the following:

$p_{c1} = \frac{1}{t}|\{\text{pixels in } B(x, y; h_n) \text{ of image 1} > c_0 \}|$,

$p_{c2} = \frac{1}{t}|\{\text{pixels in } B(x, y; h_n) \text{ of image 1} \leq c_0 \}|$,

$p_{c'1} = \frac{1}{t}|\{\text{pixels in } B(x, y; h_n) \text{ of image 2} > c_0' \}|$,

$p_{c'2} = \frac{1}{t}|\{\text{pixels in } B(x,y;h_n) \text{ of image } 2 \leq c'_0\ \}|$,

$p(1,1) =$
$\frac{1}{t}|\{\text{pixels in } B(x,y;h_n) \text{ of image } 1 > c_0\ \} \bigcap \{\text{pixels in } B(x,y;h_n) \text{ of image } 2 > c'_0\ \}|$,

$p(1,2) =$
$\frac{1}{t}|\{\text{pixels in } B(x,y;h_n) \text{ of image } 1 > c_0\ \} \bigcap \{\text{pixels in } B(x,y;h_n) \text{ of image } 2 \leq c'_0\ \}|$,

$p(2,1) =$
$\frac{1}{t}|\{\text{pixels in } B(x,y;h_n) \text{ of image } 1 \leq c_0\ \} \bigcap \{\text{pixels in } B(x,y;h_n) \text{ of image } 2 > c'_0\ \}|$,

$p(2,2) =$
$\frac{1}{t}|\{\text{pixels in } B(x,y;h_n) \text{ of image } 1 \leq c_0\ \} \bigcap \{\text{pixels in } B(x,y;h_n) \text{ of image } 2 \leq c'_0\ \}|$,

where $|.|$ denotes the cardinality of a set and t is defined as the number of pixels in the circular neighborhood $B(x,y;h_n)$. Clearly, $t = O(n^2h_n^2)$.

Then, we have the following results:

$H(\mathcal{C}(x,y)) = -p_{c1}log(p_{c1}) - p_{c2}log(p_{c2})$,

$H(\mathcal{C}'(x,y)) = -p_{c'1}log(p_{c'1}) - p_{c'2}log(p_{c'2})$,

$I(\mathcal{C}(x,y),\mathcal{C}'(x,y)) = p(1,1)log\frac{p(1,1)}{p_{c1}.p_{c'1}} + p(1,2)log\frac{p(1,2)}{p_{c1}.p_{c'2}} + p(2,1)log\frac{p(2,1)}{p_{c2}.p_{c'1}} + p(2,2)log\frac{p(2,2)}{p_{c2}.p_{c'2}}$.

As δ is the maximum mis-classification probability, under H_0,

$$
\begin{aligned}
&(a) \quad p(1,1) - p_{c1} = O(\delta) \text{ a.s.},\\
&(b) \quad p_{c1} - p_{c'1} = O(\delta) \text{ a.s.},\\
&(c) \quad p(1,2) = O(\delta) \text{ a.s.}
\end{aligned}
$$

Using (a)-(c), we have the following results:

$$
\begin{aligned}
p_{c1}log(p_{c1}) - p(1,1)log\frac{p(1,1)}{p_{c1}.p_{c'1}} &= O(\delta) \text{ a.s.},\\
p_{c2}log(p_{c2}) - p(2,2)log\frac{p(2,2)}{p_{c2}.p_{c'2}} &= O(\delta) \text{ a.s.},\\
p(1,2)log\frac{p(1,2)}{p_{c1}.p_{c'2}} &= O(\delta) \text{ a.s.},\\
p(2,1)log\frac{p(2,1)}{p_{c2}.p_{c'1}} &= O(\delta) \text{ a.s.}
\end{aligned}
$$

Therefore, under H_0, $H(\mathcal{C}(x,y)) - I(\mathcal{C}(x,y),\mathcal{C}'(x,y)) = O(\delta)$ a.s. ■

A.3.2 Proof of Proposition 4.4.4:

Note that

$$T_{VI} = \frac{1}{n^2} \sum_{i,j} D_{VI}(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j)),$$

where $D_{VI}(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j))$ is the dissimilarity metric between the clustering outcomes within $B(x_i, y_j; h_n)$ for the two images. We have

$$D_{VI}(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j)) = H(\mathcal{C}(x_i, y_j)) + H(\mathcal{C}'(x_i, y_j)) - 2I(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j)),$$

where $H(\mathcal{C}(x_i, y_j))$ and $H(\mathcal{C}'(x_i, y_j))$ are the entropy values corresponding to the images, and $I(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j))$ is the mutual information among them.

Under H_0, the true image intensity function of both images are same. Therefore, it can be shown easily that

$$H(\mathcal{C}(x_i, y_j)) - I(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j)) = O(\delta) \text{ a.s.}$$

$$H(\mathcal{C}'(x_i, y_j)) - I(\mathcal{C}(x_i, y_j), \mathcal{C}'(x_i, y_j)) = O(\delta) \text{ a.s.}$$

Hence, $T_{VI}^{H_0} = O(\delta)$ a.s. ■

Now consider the situation when H_1 is true, i.e., the true image intensity functions are not the same. The change in a neighborhood of the reference image implies a situation where the neighborhood with no clustering (i.e., neighborhood in a continuity region) transformed into a neighborhood with two clusters (i.e., neighborhood contains JLCs), neighborhood with two clusters transformed into neighborhood with no cluster. Moreover, there is also a possibility where the number of clusters is the same but cluster sizes are different. Let us define the following notations:

$\mathcal{A}_0$: regions with no change,

$\mathcal{A}_1$: regions where the continuity region is transformed into regions with two clusters,

$\mathcal{A}_2$: regions with two clusters, that are transformed into regions with continuity region,

$\mathcal{A}_3$: regions where the change is due to cluster sizes only.

Then,

$$
\begin{aligned}
T_{VI}^{H_1} &= \frac{1}{n^2}\sum_{i,j} D_{VI}(C(x_i,y_j),C'(x_i,y_j)) \\
&= \frac{1}{n^2}\Big[\sum_{\mathcal{A}_0} D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j)) + \sum_{\mathcal{A}_3} D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j)) \\
&\quad + \sum_{\mathcal{A}_1\cup\mathcal{A}_2} D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j))\Big].
\end{aligned}
$$

In $\mathcal{A}_0$, $D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j)) = O(\delta)$ a.s., since this case is similar to the situation under H_0. Also, in $\mathcal{A}_3$, i.e., the middle term of the above expression is always strictly positive by the property of a metric. Thus in $\mathcal{A}_3$, $D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j)) = O(\delta) + c_1$ a.s., for some $c_1 > 0$. Moreover, using the linearity property of the VI metric, it can be shown that for any $(i,j) \in \mathcal{A}_1 \cup \mathcal{A}_2$, $D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j)) = H(\mathcal{C}(x_i,y_j)) + H(\mathcal{C}'(x_i,y_j))$ a.s., and at least one of $H(\mathcal{C}(x_i,y_j))$ and $H(\mathcal{C}'(x_i,y_j))$ is strictly positive. Therefore, for $(i,j) \in \mathcal{A}_1 \cup \mathcal{A}_2$, $D_{VI}(\mathcal{C}(x_i,y_j),\mathcal{C}'(x_i,y_j)) = c_2$ a.s., where $c_2 > 0$. Hence, combining all results, $T_{VI}^{H_1} = O(\delta) + c$ a.s. for some $c > 0$. ■

A.4 Sketches of Proofs of the Stated Propositions in Chapter 5

Lemma A.4.1. *Under the assumption mentioned in Proposition 5.3.1 in Section 5.3,*

$$D_H(E \cap \Omega_{\bar{S},h_n}, \widehat{E} \cap \Omega_{\bar{S},h_n}) = O(h_n), \ a.s.,$$

where $D_H(A,B)$ is the Hausdorff distance between two point-sets A and B.

Proof of Lemma A.4.1: See Theorem 1 of Kang and Qiu (2014) for the proof of Lemma A.4.1. This lemma demonstrates the strong consistency of the detected edge pixels using the LLK-based jump location curve estimation procedure described in Section 5.2.1. ■

Intuitively, it is evident from the proof of Lemma A.4.1 that any pixel within a circular neighborhood of h_n of the true edge pixel is detected as edge pixel with probability one when the image resolution is large, i.e., $n \to \infty$, $\mathbb{P}\big[(x_i,y_j) \text{ is detected as edge pixel}\big] = 1$, if $D((x,y),(x_i,y_j)) \le h_n$ for $(x,y) \in E \cap \Omega_{\bar{S},h_n}$.

Similarly, $\mathbb{P}\big[(x_i,y_j) \text{ is NOT detected as edge pixel}\big] = 1$, if $D((x,y),(x_i,y_j)) > h_n$ for $(x,y) \in E \cap \Omega_{\bar{S},h_n}$.

A.4.1 Proof of Proposition 5.3.1:

From Lemma A.4.1, we know that $D_H(E \cap \Omega_{\bar{S},h_n}, \widehat{E} \cap \Omega_{\bar{S},h_n}) = O(h_n)$ *a.s.*

Let us define $U = E \cap \Omega_{\bar{S},h_n}$ and $\widehat{U} = \widehat{E} \cap \Omega_{\bar{S},h_n}$. Then we have,

$$D_H(U, \widehat{U}) = \max\{\delta(U, \widehat{U}), \delta(\widehat{U}, U)\}, \quad \ldots\ldots\ldots \quad (i)$$

where $\delta(U, \widehat{U}) = \sup_{(x,y)\in U} \inf_{(x',y')\in \widehat{U}} D((x,y),(x',y'))$.
Therefore, from (i), we have

$$\delta(U, \widehat{U}) \leq O(h_n), \ a.s., \quad \ldots\ldots\ldots \quad (ii)$$

$$\delta(\widehat{U}, U) \leq O(h_n), \ a.s. \quad \ldots\ldots\ldots \quad (iii)$$

From (ii),

$$\sup_{(x,y)\in U} \inf_{(x',y')\in \widehat{U}} D((x,y),(x',y')) \leq O(h_n) \ a.s.$$
$$\text{i.e.,} \inf_{(x',y')\in \widehat{U}} D((x,y),(x',y')) \leq O(h_n) \ a.s., \quad \forall (x,y) \in U.$$

Therefore, given any $(x,y) \in U$, $\exists (x',y') \in \widehat{U}$ such that $D((x,y),(x',y')) \leq O(h_n), \ a.s.$
Similarly, using (iii), given any $(x',y') \in \widehat{U}$, $\exists (x,y) \in U$ such that $D((x',y'),(x,y)) \leq O(h_n)$ *a.s.*
Hence,

$$\widehat{\mathcal{C}} = \frac{\sum_{i,j=1}^{n} (x'_i, y'_j) \mathbb{I}_{(x'_i,y'_j)}(\widehat{U})}{|\widehat{U}|} = \frac{\sum_{i,j=1}^{n} (x_i, y_j) \mathbb{I}_{(x'_i,y'_j)}(\widehat{U})}{|\widehat{U}|} + O(h_n) \ a.s. \quad \ldots\ldots \quad (iv)$$

Here, we replace each $(x'_i, y'_j) \in \widehat{U}$ by $(x_i, y_j) + O(h_n)$, where (x_i, y_j) is the nearest true edge pixel in U. Due to Lemma A.4.1, if a non-singular pixel is within h_n-neighborhood of a JLC, it will be detected as an edge pixel almost surely. Hence, the detected edge pixels around the true JLC form a band of width $2h_n = \frac{1}{n}(2nh_n)$. Note that $\frac{1}{n}$ is the distance between two nearest pixels. Hence, for large n, the number of detected edge pixels for which we get the same closest true edge pixel is $(2nh_n - o(nh_n))$. We subtract $o(nh_n)$ because the asymptotic probability that we fail to detect a pixel as an edge pixel within h_n of a true JLC is zero. Consequently,

$$\frac{|\widehat{U}|}{2nh_n|U|} \xrightarrow{a.s.} 1 \quad \ldots\ldots\ldots \quad (v)$$

Therefore, for large n,

$$\frac{\sum_{i,j=1}^{n}(x_i,y_j)\mathbb{I}_{(x_i',y_j')}(\widehat{U})}{|\widehat{U}|}=\frac{(2nh_n-o(nh_n))\sum_{i,j=1}^{n}(x_i,y_j)\mathbb{I}_{(x_i,y_j)}(U)}{|\widehat{U}|}. \quad \ldots\ldots\ldots \quad (vi)$$

From (iv), (v), and (vi), $\widehat{\mathcal{C}} \xrightarrow{a.s.} |U|^{-1}\sum_{i,j=1}^{n}(x_i,y_j)\mathbb{I}_{(x_i,y_j)}(U)=\mathcal{C}$.
Hence, Proposition 5.3.1 is proved. ■

Proof of Corollary 5.3.2:
From Proposition 5.3.1, it is clear that, for a detected jump point $(x_i,y_j)\in\widehat{U}$ there exists at least one (x,y) such that $D((x_i,y_j),(x,y))=O(h_n)$ $a.s.$ Then,

$$\begin{aligned}D((x_i,y_j),\widehat{\mathcal{C}}) &\le D((x_i,y_j),(x,y))+D((x,y),\widehat{\mathcal{C}})\\ &\le O(h_n)+D((x,y),\mathcal{C})+D(\mathcal{C},\widehat{\mathcal{C}})\ a.s.\\ &= D((x,y),\mathcal{C})+O(h_n)\ a.s.\end{aligned}$$

Similarly, $D((x,y),\mathcal{C})\le D((x_i,y_j),\widehat{\mathcal{C}})+O(h_n)$ $a.s.$
Therefore, $|D((x_i,y_j),\widehat{\mathcal{C}})-D((x,y),\mathcal{C})|=O(h_n)$ $a.s.$ ■

Proof of Corollary 5.3.3:
Let R_{Dn} be one sample radial distance from $\widetilde{F}$. Then,

$$\begin{aligned}\widetilde{F}(z) &= \mathbb{P}(R_{Dn}\le z)\\ &= \mathbb{P}(R_D+O(h_n)\le z)\ a.s. \quad \text{(from Corollary 5.3.2)}\\ &= F(z+O(h_n))\ a.s.\end{aligned}$$

Therefore, under the assumption that the distribution of the radial distance is absolutely continuous, for large n, $\widetilde{F}(z)\xrightarrow{a.s.}F(z)$. ■

A.4.2 Proof of Proposition 5.3.4:

From Corollary 5.3.3, it is clear that

$$\widetilde{F}_1(z)\xrightarrow{a.s.}F_1(z),$$

$$\widetilde{F}_2(z)\xrightarrow{a.s.}F_2(z).$$

Note that,

$$T_{CVM}(\widetilde{F}_1,\widetilde{F}_2)=\frac{r_1r_2}{r_1+r_2}\int_{-\infty}^{\infty}[\widetilde{F}_1(x)-\widetilde{F}_2(x)]^2d\widetilde{F}_c(x)$$

Note that, $\widetilde{F}_1(z)$ and $\widetilde{F}_2(z)$ are the plug in estimator of F_1 and F_2, respectively. From Corollary 5.3.3, we have shown that $\widetilde{F}_1(z)$ and $\widetilde{F}_2(z)$ are consistent estimates of the distribution from which the samples are drawn. Therefore, the procedure is consistent, i.e., $T_{CVM}(\widetilde{F}_1, \widetilde{F}_2) \xrightarrow{a.s.} T_{CVM}(F_1, F_2)$. The proof of the proposition follows from Curry et al. (2019), and Anderson (1962). ■

A.4.3 Theoretical Justification of Bootstrapping in Section 5.2.2

Here we investigate statistical consistency to substantiate the proposed bootstrap algorithm in Section 5.2.2. Before going into details, we introduce relevant notations below. The distribution of the test statistic T_{CVM} is denoted by $\mathcal{F}(x)$, and its empirical distribution is denoted by $\widehat{\mathcal{F}}(x)$. We denote its bootstrap distribution by $\widehat{\mathcal{F}}^*(x)$ which can be expressed as

$$\widehat{\mathcal{F}}^*(x) = \frac{1}{B}\sum \mathbb{I}_{\{T^*_{CVM}(\widetilde{F}_1,\widetilde{F}_2)\leq x\}},$$

where B is the number of bootstrap samples and $T^*_{CVM}(\widetilde{F}_1, \widetilde{F}_2)$ is the computed Cramér-von Mises distance with the bootstrapped image samples.

We define the shape difference between two images $\mathcal{I}_1$ and $\mathcal{I}_2$ by $d_{GH}(\mathcal{I}_1, \mathcal{I}_2)$, the *Gromov-Hausdorff* distance which is invariant under rigid-body transformation. Now under H_0, $\mathcal{I}_1 = \mathcal{I}_0 + E_1$ and $\mathcal{I}_2 = R\mathcal{I}_0 + E_2$, where R is a rigid-body transformation and $E_k (k = 1, 2)$ is the pointwise noise in the two images. Define a transformation $\mathcal{T} : \mathbb{R}^{n\times n} \to \mathbb{R}$ such that

$$\mathcal{T}(\mathcal{I}_1, \mathcal{I}_2) = T_{CVM}(\widetilde{F}_1, \widetilde{F}_2).$$

As radial distance is invariant under rotation and translation on the image object, we have $\mathcal{T}(\mathcal{I}_0, R\mathcal{I}_0) = 0$. Moreover, for a small changes in the shape of the image object the Cramér-von Mises distance will change a little only. Let us assume that $\mathcal{T}$ is continuous under the metric $d_{GH}(\quad)$ defined earlier. Thus, given any $\epsilon > 0$, there exists $\delta > 0$ such that $d_{GH}(\mathcal{I}_1, \mathcal{I}^*_1) \leq \delta$,

i.e., $|\mathcal{T}(\mathcal{I}_1, \mathcal{I}_2) - \mathcal{T}(\mathcal{I}^*_1, \mathcal{I}^*_2)| < \epsilon$
i.e., $|\mathcal{T}(\mathcal{I}_0 + E_1, R\mathcal{I}_0 + E_2) - \mathcal{T}(\widehat{\mathcal{I}}_0 + E^*_1, R\widehat{\mathcal{I}}_0 + E^*_2)| < \epsilon$
i.e., $\mathcal{T}(\mathcal{I}_0 + E_1, R\mathcal{I}_0 + E_2) - \epsilon < \mathcal{T}(\widehat{\mathcal{I}}_0 + E^*_1, R\widehat{\mathcal{I}}_0 + E^*_2) < \mathcal{T}(\mathcal{I}_0 + E_1, R\mathcal{I}_0 + E_2) + \epsilon$.
Here E^*_k is the added residual can be taken with replacement from the set $\{\mathcal{I}_k - \widehat{I}_k\}$, $k = 1, 2$. Hence,

$\frac{1}{B}\sum \mathbb{I}_{\{\mathcal{T}(\mathcal{I}_0+E_1,R\mathcal{I}_0+E_2)\leq x-\epsilon\}} < \frac{1}{B}\sum \mathbb{I}_{\{\mathcal{T}(\widehat{\mathcal{I}}_0+E^*_1,R\widehat{\mathcal{I}}_0+E^*_2)\leq x\}} < \frac{1}{B}\sum \mathbb{I}_{\{\mathcal{T}(\mathcal{I}_0+E_1,R\mathcal{I}_0+E_2)\leq x+\epsilon\}}$
i.e., $\frac{1}{B}\sum \mathbb{I}_{\{T_{CVM}(\widetilde{F}_1,\widetilde{F}_2)\leq x-\epsilon\}} < \frac{1}{B}\sum \mathbb{I}_{T^*_{CVM}(\widetilde{F}_1,\widetilde{F}_2)\leq x\}} < \frac{1}{B}\sum \mathbb{I}_{\{T_{CVM}(\widetilde{F}_1,\widetilde{F}_2)\leq x+\epsilon\}}$
i.e., $\widehat{\mathcal{F}}(x - \epsilon) < \widehat{\mathcal{F}}^*(x) < \widehat{\mathcal{F}}(x + \epsilon)$.

We are now ready to sketch the proof of consistency of $\widehat{\mathcal{F}}^*(x)$ under Kolmogorov metric. We start with the definition of bootstrap consistency. By definition, $\widehat{\mathcal{F}}^*$ is strongly consistent of $\widehat{\mathcal{F}}$ under Kolmogorov metric if $K(\widehat{\mathcal{F}}^*, \widehat{\mathcal{F}}) \to 0 \quad a.s.$, where $K()$ denotes Kolmogorov metric. Now we can write

$$K(\widehat{\mathcal{F}}^*, \widehat{\mathcal{F}}) = \sup_x |\widehat{\mathcal{F}}^*(x) - \widehat{\mathcal{F}}(x)| \leq \sup_x |\widehat{\mathcal{F}}^*(x) - \mathcal{F}(x)| + \sup_x |\widehat{\mathcal{F}}^(x) - \mathcal{F}(x)|.$$

By Glivenko Cantelli Theorem, $\sup_x |\widehat{\mathcal{F}}(x) - \mathcal{F}(x)| \to 0 \quad a.s.$, as the image resolution increases. Moreover,
$\sup_x |\widehat{\mathcal{F}}(x-\epsilon) - \mathcal{F}(x)| \leq \sup_x |\widehat{\mathcal{F}}(x-\epsilon) - \mathcal{F}(x-\epsilon)| + \sup_x |\mathcal{F}(x-\epsilon) - \mathcal{F}(x)|$.
Since $\mathcal{F}(x)$ is continuous, $\sup_x |\mathcal{F}(x-\epsilon) - \mathcal{F}(x)| \to 0$ and again by Glivenko Cantelli Theorem, $\sup_x |\widehat{\mathcal{F}}(x-\epsilon) - \mathcal{F}(x-\epsilon)| \to 0 \quad a.s.$ Hence, the proposed bootstrap is strongly consistent under Kolmogorov metric. ■

Bibliography

Amirkhani, F. and Amiri, A. (2020). A novel framework for spatiotemporal monitoring and post-signal diagnosis of processes with image data. *Quality and Reliability Engineering International*, 36(2):705–735.

Anderson, T. W. (1962). On the distribution of the two-sample cramer-von mises criterion. *The Annals of Mathematical Statistics*, pages 1148–1159.

Anderson, T. W. and Darling, D. A. (1954). A test of goodness of fit. *Journal of the American statistical association*, 49(268):765–769.

Avants, B., Epstein, C., Grossman, M., and Gee, J. (2008). Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12(1):26–41. Special Issue on The Third International Workshop on Biomedical Image Registration – WBIR 2006.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 48(3):259–279.

Bowman, A. W. (2006). Comparing nonparametric surfaces. *Statistical Modelling*, 6(4):279–299.

Bui, A. T. and Apley, D. W. (2018a). A monitoring and diagnostic approach for stochastic textured surfaces. *Technometrics*, 60(1):1–13.

Bui, A. T. and Apley, D. W. (2018b). *A Monitoring and Diagnostic Approach for Stochastic Textured Surfaces*.

Bui, A. T. and Apley, D. W. (2021). spc4sts: Statistical process control for stochastic textured surfaces in r. *Journal of Quality Technology*, 53(3):219–242.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Chakraborti, S., Human, S., and Graham, M. (2008). Phase i statistical process control charts: An overview and some results. *Quality Engineering*, 21(1):52–62.

Chen, Y., Birch, J. B., and Woodall, W. H. (2015). Cluster-based profile analysis in phase *I. Journal of Quality Technology*, 47(1):14–29.

Clark, J. (1988). Singularity theory and phantom edges in scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):720–727.

Clayden, J., Modat, M., Presles, B., Anthopoulos, T., and Daga, P. (2023). *RNiftyReg: Image Registration Using the 'NiftyReg' Library*. R package version 2.8.1.

Curry, J., Dang, X., and Sang, H. (2019). A rank-based cramér–von-mises-type test for two samples. *Brazilian Journal of Probability and Statistics*, 33(3):425–454.

Dai, X. and Müller, H.-G. (2018). Principal component analysis for functional data on riemannian manifolds and spheres. *The Annals of Statistics*.

Das, S., Roy, A., and Mukherjee, P. S. (2024). Image registration for zooming: A statistically consistent local feature mapping approach. *Stat*, 13(1):e664.

Davatzikos, C., Prince, J. L., and Bryan, R. N. (1996). Image registration based on boundary mapping. *IEEE Transactions on medical imaging*, 15(1):112–115.

Dette, H. and Munk, A. (1998a). Nonparametric comparison of several regression functions: exact and asymptotic theory. *The Annals of Statistics*, 26(6):2339–2368.

Dette, H. and Munk, A. (1998b). Testing heteroscedasticity in nonparametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(4):693–708.

Dette, H. and Neumeyer, N. (2001). Nonparametric analysis of covariance. *the Annals of Statistics*, 29(5):1361–1400.

Dong, H., Chen, N., and Wang, K. (2020). Modeling and change detection for count-weighted multilayer networks. *Technometrics*, 62(2):184–195.

Feng, L. and Qiu, P. (2018). Difference detection between two images for image monitoring. *Technometrics*, 60(3):345–359.

Fessler, J. A., Erdogan, H., and Wu, W. B. (2000). Exact distribution of edge-preserving map estimators for linear signal models with gaussian measurement noise. *IEEE Transactions on image processing*, 9(6):1049–1055.

Figueiredo, M. A. and Nowak, R. D. (2003). An em algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916.

Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-6(6):721–741.

Godtliebsen, F. and Sebastiani, G. (1994). Statistical methods for noisy images with discontinuities. *Journal of Applied Statistics*, 21(5):459–477.

Gonzalez, R. C. and Woods, R. E. (2019). *Digital Image Processing*. Pearson Education.

Grasso, M. and Colosimo, B. M. (2017). Process defects and in situ monitoring methods in metal powder bed fusion: a review. *Measurement Science and Technology*, 28(4):044005.

Hall, P. and Hart, J. D. (1990). Bootstrap test for difference between means in nonparametric regression. *Journal of the American Statistical Association*, 85(412):1039–1049.

Hall, P. and Qiu, P. (2007a). Blind deconvolution and deblurring in image analysis. *Statistica Sinica*, pages 1483–1509.

Hall, P. and Qiu, P. (2007b). Nonparametric estimation of a point-spread function in multivariate problems. *The Annals of Statistics*, 35(4).

Hawkins, D. M. (1987). Self-starting cusum charts for location and scale. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 36(4):299–316.

Hawkins, D. M. and Deng, Q. (2010). A nonparametric change-point control chart. *Journal of Quality Technology*, 42(2):165–173.

Hawkins, D. M. and Olwell, D. H. (2012). *Cumulative sum charts and charting for quality improvement*. Springer Science & Business Media.

Hawkins, D. M., Qiu, P., and Kang, C. W. (2003). The changepoint model for statistical process control. *Journal of quality technology*, 35(4):355–366.

He, Z., Zuo, L., Zhang, M., and Megahed, F. M. (2016). An image-based multivariate generalized likelihood ratio control chart for detecting and diagnosing multiple faults in manufactured products. *International Journal of Production Research*, 54(6):1771–1784.

Holland, M. D. and Hawkins, D. M. (2014). A control chart based on a nonparametric multivariate change-point model. *Journal of Quality Technology*, 46(1):63–77.

Jones, B. A. and Fleck, J. (2020). Shrinking lakes, air pollution, and human health: Evidence from california's salton sea. *Science of the Total Environment*, 712:136490.

Joo, J.-H. and Qiu, P. (2009). Jump detection in a regression curve and its derivative. *Technometrics*, 51(3):289–305.

Joshi, M. V. and Chaudhuri, S. (2005). Joint blind restoration and surface recovery in photometric stereo. *JOSA A*, 22(6):1066–1076.

Kang, Y. (2020). Consistent blind image deblurring using jump-preserving extrapolation. *Journal of Computational and Graphical Statistics*, 29(2):372–382.

Kang, Y. (2022). Statistical quality control using image intelligence: A sparse learning approach. *Naval Research Logistics (NRL)*, 69(7):996–1008.

Kang, Y. (2023). *DRIP: Discontinuous Regression and Image Processing*. R package version 1.8.

Kang, Y., Mukherjee, P. S., and Qiu, P. (2018). Efficient blind image deblurring using nonparametric regression and local pixel clustering. *Technometrics*, 60(4):522–531.

Kang, Y. and Qiu, P. (2014). Jump detection in blurred regression surfaces. *Technometrics*, 56(4):539–550.

King, E., Hart, J. D., and Wehrly, T. E. (1991). Testing the equality of two regression curves using linear smoothers. *Statistics & Probability Letters*, 12(3):239–247.

Koosha, M., Noorossana, R., and Megahed, F. (2017). Statistical process monitoring via image data using wavelets. *Quality and Reliability Engineering International*, 33(8):2059–2073.

Kulasekera, K. (1995). Comparison of regression curves using quasi-residuals. *Journal of the American Statistical Association*, 90(431):1085–1093.

Lorden, G. (1971). Procedures for reacting to a change in distribution. *The annals of mathematical statistics*, pages 1897–1908.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110.

Mammen, E. (1993). Bootstrap and wild bootstrap for high dimensional linear models. *The annals of statistics*, 21(1):255–285.

Marroquin, J. L., Velasco, F. A., Rivera, M., and Nakamura, M. (2001). Gauss-markov measure field models for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):337–348.

Megahed, F., Wells, L., and Camelio, J. (2010). The use of 3d laser scanners in statistical process control. Technical report, SAE Technical Paper.

Megahed, F. M., Wells, L. J., Camelio, J. A., and Woodall, W. H. (2012). A spatiotemporal method for the monitoring of image data. *Quality and Reliability Engineering International*, 28(8):967–980.

Megahed, F. M., Woodall, W. H., and Camelio, J. A. (2011). A review and perspective on control charting with image data. *Journal of Quality Technology*, 43(2):83–98.

Mei, Y. (2010). Efficient scalable schemes for monitoring a large number of data streams. *Biometrika*, 97(2):419–433.

Mei, Y. (2011). Quickest detection in censoring sensor networks. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 2148–2152.

Meilă, M. and Heckerman, D. (2001). An experimental comparison of model-based clustering methods. *Machine learning*, 42:9–29.

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Montgomery, D. C. (2007). *Introduction to statistical quality control*. John Wiley & Sons.

Moustakides, G. V. (1986). Optimal stopping times for detecting changes in distributions. *the Annals of Statistics*, 14(4):1379–1387.

Mukherjee, P. S. (2023). Change-point-based statistical process controls. In *Springer Handbook of Engineering Statistics*, pages 361–381. Springer.

Mukherjee, P. S. and Qiu, P. (2011). 3-d image denoising by local smoothing and nonparametric regression. *Technometrics*, 53(2):196–208.

Mukherjee, P. S. and Qiu, P. (2013). Efficient bias correction for magnetic resonance image denoising. *Statistics in medicine*, 32(12):2079–2096.

Mukherjee, P. S. and Qiu, P. (2015). Image denoising by a local clustering framework. *Journal of Computational and Graphical Statistics*, 24(1):254–273.

Okhrin, Y., Schmid, W., and Semeniuk, I. (2019). Monitoring image processes: overview and comparison study. In *International workshop on intelligent statistical quality control*, pages 143–163. Springer.

Okhrin, Y., Schmid, W., and Semeniuk, I. (2020). New approaches for monitoring image data. *IEEE Transactions on Image Processing*, 30:921–933.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Otto, P. (2019). Parallelized monitoring of dependent spatiotemporal processes. In *International Workshop on Intelligent Statistical Quality Control*, pages 165–183. Springer.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.

Polzehl, J. and Spokoiny, V. G. (2000). Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):335–354.

Qiu, P. (1998). Discontinuous regression surfaces fitting. *The Annals of Statistics*, 26(6):2218–2245.

Qiu, P. (2004). The local piecewisely linear kernel smoothing procedure for fitting jump regression surfaces. *Technometrics*, 46(1):87–98.

Qiu, P. (2005). *Image processing and jump regression analysis*. John Wiley & Sons.

Qiu, P. (2007). Jump surface estimation, edge detection, and image restoration. *Journal of the American Statistical Association*, 102(478):745–756.

Qiu, P. (2009). Jump-preserving surface reconstruction from noisy data. *Annals of the Institute of Statistical Mathematics*, 61(3):715–751.

Qiu, P. (2013). *Introduction to statistical process control*. CRC press.

Qiu, P. (2018). Jump regression, image processing, and quality control. *Quality Engineering*, 30(1):137–153.

Qiu, P. (2019). Some recent studies in statistical process control. *Statistical quality technologies: theory and practice*, pages 3–19.

Qiu, P. (2024). *Statistical Methods for Dynamic Disease Screening and Spatio-Temporal Disease Surveillance*. CRC Press.

Qiu, P. and Bhandarkar, S. M. (1996). An edge detection technique using local smoothing and statistical hypothesis testing. *Pattern Recognition Letters*, 17(8):849–872.

Qiu, P. and Kang, Y. (2015). Blind image deblurring using jump regression analysis. *Statistica Sinica*, pages 879–899.

Qiu, P. and Mukherjee, P. S. (2010). Edge structure preserving image denoising. *Signal Processing*, 90(10):2851–2862.

Qiu, P. and Mukherjee, P. S. (2011). Edge structure preserving 3d image denoising by local surface approximation. *IEEE transactions on pattern analysis and machine intelligence*, 34(8):1457–1468.

Qiu, P. and Xing, C. (2013). On nonparametric image registration. *Technometrics*, 55(2):174–188.

Qiu, P. and Yandell, B. (1997). Jump detection in regression surfaces. *Journal of Computational and Graphical Statistics*, 6(3):332–354.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Reddy, B. S. and Chatterji, B. N. (1996). An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*, 5(8):1266–1271.

Reynolds, M. R. (1975). Approximations to the average run length in cumulative sum control charts. *Technometrics*, 17(1):65–71.

Rivera, M. and Marroquin, J. L. (2002). Adaptive rest condition potentials: first and second order edge-preserving regularization. *Computer Vision and Image Understanding*, 88(2):76–93.

Roberts, S. (1959). Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–101.

Rosenblatt, M. (1952). Limit theorems associated with variants of the von mises statistic. *The Annals of Mathematical Statistics*, pages 617–623.

Roy, A. and Mukherjee, P. S. (2024a). A control chart for monitoring images using jump location curves. *Quality Engineering*, 36(2):439–452.

Roy, A. and Mukherjee, P. S. (2024b). Image comparison based on local pixel clustering. *Technometrics*, 66(4):495–506.

Shewhart, W. A. (1931). *Economic control of quality of manufactured product*. Macmillan And Co Ltd, London.

Shuford, W. D., Warnock, N., Molina, K. C., and Sturm, K. K. (2002). The salton sea as critical habitat to migratory and resident waterbirds. *Hydrobiologia*, 473:255–274.

Siddique, N., Paheding, S., Elkin, C. P., and Devabhaktuni, V. (2021). U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE access*, 9:82031–82057.

Stamatakos, G. and Giatili, S. (2017). A numerical handling of the boundary conditions imposed by the skull on an inhomogeneous diffusionreaction model of glioblastoma invasion into the brain: Clinical validation aspects. *Cancer informatics*, 16:1–16.

Stevens, N. T., Wilson, J. D., Driscoll, A. R., McCulloh, I., Michailidis, G., Paris, C., Parker, P., Paynabar, K., Perry, M. B., Reisi-Gahrooei, M., et al. (2021a). Research in network monitoring: Connections with spm and new directions. *Quality Engineering*, 33(4):736–748.

Stevens, N. T., Wilson, J. D., Driscoll, A. R., McCulloh, I., Michailidis, G., Paris, C., Paynabar, K., Perry, M. B., Reisi-Gahrooei, M., Sengupta, S., et al. (2021b). Foundations of network monitoring: Definitions and applications. *Quality Engineering*, 33(4):719–730.

Su, J., Kurtek, S., Klassen, E., and Srivastava, A. (2014). Statistical analysis of trajectories on riemannian manifolds: bird migration, hurricane tracking and video surveillance. *The Annals of Applied Statistics*.

Sun, J. and Qiu, P. (2007). Jump detection in regression surfaces using both first-order and second-order derivatives. *Journal of Computational and Graphical Statistics*, 16(2):289–311.

Torre, V. and Poggio, T. A. (1986). On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):147–163.

Vining, G. (2009). Technical advice: Phase i and phase ii control charts. *Quality Engineering*, 21(4):478–479.

Wang, X.-F. and Ye, D. (2010). On nonparametric comparison of images and regression surfaces. *Journal of statistical planning and inference*, 140(10):2875–2884.

Wolberg, G. and Zokai, S. (2000). Robust image registration using log-polar transform. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 493–496. IEEE.

Woodall, W. H. and Ncube, M. M. (1985). Multivariate cusum quality-control procedures. *Technometrics*, 27(3):285–292.

Xing, C. and Qiu, P. (2011). Intensity-based image registration by nonparametric local smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2081–2092.

Yan, H., Paynabar, K., and Shi, J. (2017). Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics*, 59(1):102–114.

Yi, F. and Qiu, P. (2021). Edge-preserving denoising of image sequences. *Entropy*, 23(10):1332.

Yi, F. and Qiu, P. (2023). Water resource surveillance for the salton sea in california by adaptive sequential monitoring of its landsat images. *Journal of Agricultural, Biological and Environmental Statistics*, pages 1–15.

Zamba, K. and Hawkins, D. M. (2006). A multivariate change-point model for statistical process control. *Technometrics*, 48(4):539–549.

Zang, Y. and Qiu, P. (2018a). Phase *I* monitoring of spatial surface data from 3d printing. *Technometrics*, 60(2):169–180.

Zang, Y. and Qiu, P. (2018b). Phase *II* monitoring of free-form surfaces: An application to 3d printing. *Journal of Quality Technology*, 50(4):379–390.

Zhang, X., Gilliam, C., and Blu, T. (2020). All-pass parametric image registration. *IEEE Transactions on Image Processing*, 29:5625–5640.

Zhao, S., Bakoyannis, G., Lourens, S., and Tu, W. (2020). Comparison of nonlinear curves and surfaces. *Computational Statistics & Data Analysis*, 150:106987.

Zou, C. and Qiu, P. (2009). Multivariate statistical process control using lasso. *Journal of the American Statistical Association*, 104(488):1586–1596.

Zou, C. and Tsung, F. (2010). Likelihood ratio-based distribution-free ewma control charts. *Journal of Quality Technology*, 42(2):174–196.

Zou, C. and Tsung, F. (2011). A multivariate sign ewma control chart. *Technometrics*, 53(1):84–97.

Zwetsloot, I. and Woodall, W. (2021). A review of some sampling and aggregation strategies for basic statistical process monitoring. *Journal of Quality Technology*, 53:1–16.

Zwetsloot, I. M., Jones-Farmer, L. A., and Woodall, W. H. (2024). Monitoring univariate processes using control charts: Some practical issues and advice. *Quality Engineering*, 36(3):487–499.