NORTHWESTERN UNIVERSITY

Statistical Process Control of Stochastic Textured Surfaces

A DISSERTATION

SUMBITTED TO THE GRADUATE SCHOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Industrial Engineering and Management Sciences

BY

Anh Tuan Bui

EVANSTON, ILLINOIS

September 2019

ABSTRACT

Statistical Process Control of Stochastic Textured Surfaces

Anh Tuan Bui

This dissertation develops a new framework and algorithms for statistical process control of stochastic textured surface data that have no distinct features other than stochastic characteristics that vary randomly (e.g., image data of textiles or material microstructures and surface metrology data of metal parts). All methods are general and nonparametric in that they require no prior knowledge of the types of abnormalities that might occur nor the extraction of specific predefined features. The methods are applicable to a wide range of materials and address unsolved problems regarding monitoring and diagnosing quality-related issues that can lead to early damage, reduced lifetime, or compromised aesthetics of the manufactured materials. Specifically, the first problem is detecting defects on the surfaces (e.g., microstructure porosities); the second problem is detecting changes that affect the entire nature of the surface textures (e.g., microstructure morphology changes); and the third problem is characterizing previously unidentified sample-to-sample variation in the surface textures, in a manner that is conducive to conveying an understanding of the physical nature of the variation. To solve these problems, we use supervised learning methods to model the stochastic behavior of the stochastic textured surface samples. For local defects, we propose two spatial moving statistics for detecting local aberrations in the textured surfaces, based on the residuals of the supervised learning model (fitted to an in-control sample) applied to new samples. For global changes, we develop a monitoring statistic using likelihood-ratio principles to detect changes in the surface nature, relative to the in-control one. For understanding surface variation, we derive dissimilarity measures between surface samples and use manifold learning on these dissimilarities to discover a low-dimensional parameterization of the surface variation patterns. Visualizing how the surfaces change as the manifold parameters are varied helps build an understanding of the physical characteristic of each variation pattern.

ACKNOWLEDGEMENT

This dissertation materializes from the supports of many organizations and people that I have been blessed to have along my academic journey. First, I would like to acknowledge the supports from the National Science Foundation, the Air Force Office of Scientific Research, the Vietnam Education Foundation, the American Statistical Association, and Northwestern University and its benefactors. Next, I would like to thank my advisor, Dr. Apley, for the invaluable advices and direct contributions as a coauthor of the works in this dissertation. I also would like to thank my dissertation committee members, Dr. Chen and Dr. Malthouse, and the anonymous journal reviewers for the manuscripts of my dissertation works for their helpful suggestions. Furthermore, I would like to thank the many professors and teachers from whom I have learned much knowledge, the many friends together with whom I have had much fun, and the many relatives from whom I have gained much help, even though I cannot list all their names here. Especially, I would like to thank my parents-in-law, Hanh and Tam, for being tremendously supportive. Last but not least, I would like to thank my wife, Hoa, for her love and sacrifices and being with me always; my son, An, for the joy like no other; and my parents, Thao and Son, for raising me through many difficulties to become who I am today. Thank you all for making this happens.

To my wife and son, Hoa and An,

and my parents, Thao and Son.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

This dissertation is a collection of three papers (Bui and Apley 2018a, 2018b, 2019a) that develop a new framework and algorithms for statistical process control (SPC) of stochastic textured surface data. Such data do not have distinct features other than stochastic characteristics that vary randomly. Some examples include image data of textiles that show weave patterns or material microstructures. Point cloud surface roughness data of machined, cast, or formed metal parts, obtained from either a contact stylus trace or optical laser scanning, is another example of the stochastic textured surface data. However, throughout this dissertation, we illustrate the approaches with image data. Section 2.1 will explain this type of data and its challenges for SPC purposes in detail. All methods in this dissertation are general and nonparametric in that they require no prior knowledge of the types of abnormalities that might occur nor the extraction of specific predefined features. They are applicable to a wide range of materials and address unsolved problems regarding monitoring and diagnosing quality-related issues that can lead to damage, reduced lifetime, or compromising aesthetics of the manufactured materials.

Specifically, in Chapter 2 (Bui and Apley 2018a) and Chapter 3 (Bui and Apley 2018b), we develop two general approaches that can detect general deviations from the normal in-control statistical behavior of the stochastic textured surfaces, where the normal in-control statistical behavior is modeled in a reasonably generic manner from the in-control training images. In both of these works, we do not assume any prior knowledge of the defects nor must defects be persistent across different images. In this regard, the approaches are analogous to the classic Shewhart control charting approach (Montomery 2009), which generically characterizes in-control behavior from a training sample (Phase I), and monitors future samples to detect general departures from the in-control behavior (Phase II).

The method in Chapter 2 is developed to monitor for *local defects*, which are essentially

deviations from the normal statistical behavior that occur only at some local areas of the stochastic textured surfaces. We use an off-the-shelf supervised learning algorithm to characterize the in-control statistical behavior of the stochastic texture surfaces of interest in a generic manner. Our primary monitoring statistic is derived from some spatial moving statistic, computed from the residual errors of the supervised learning model. In addition to monitoring, our approach is designed to help users diagnose the cause of the deviations from the normal behavior via highlighting pixels with large spatial moving statistics.

In Chapter 3, our objective is to monitor for another class of texture-related defects that involves changes in the nature of the entire stochastic textured surface as opposed to the local defects. We refer to this class of changes as *global changes*. Again, we use supervised learning to estimate the joint distribution that completely represents the statistical behavior of the stochastic texture surfaces. Because a global change would result in a change in the joint distribution, we develop a monitoring statistic based on the generalized likelihood-ratio test (GLRT) principle for detecting such a deviation.

Unlike the problems in Chapters 2 and 3, which are mainly about monitoring for future deviations from the normal in-control behavior, the problem in Chapter 4 is completely different. In this work, we focus on a more exploratory diagnostic objective of understanding the nature of the variation across a set of stochastic textured surface data samples, given that their stochastic behavior is not stable and varies across the set. To this end, we derive two new pairwise dissimilarity measures for the stochastic textured surfaces based on a novel application of the Kullback-Leibler divergence. Then, we use a form of manifold learning applied to the pairwise dissimilarities of the given set of samples. The learned manifold coordinates provide a parameterization of the variation existing in the set of samples, which can then be visualized (as will be described later) to reveal the nature of the variation. This helps us understand the sources of these variations in the current manufacturing process, and aid the process improvement. Hence,

the approach in Chapter 4 is purely for diagnostic purposes.

All the computer codes used in this dissertation have been released in the **spc4sts** R package. The help files, examples, and other documents accompanying this package are useful resources aimed at helping practitioners understand the technical details and effectively apply the approaches. See Bui and Apley (2019b) for an introduction of the **spc4sts** package.

CHAPTER 2

Local Defect Monitoring and Diagnostics for Stochastic Textured Surfaces

2.1 Introduction

Image and other profile data are increasingly commonly collected for manufacturing quality control purposes. We consider a subcategory of such data that we refer to as stochastic textured surface data, which can be viewed as 2-D stochastic processes. For example, Figure 2.1(a) is an image of a textile material with enough magnification to show the weave patterns, which exhibit a great deal of stochastic behavior and are not deterministically positioned. Figure 2.1(b) is a greyscale image version of a simulated 2-D stochastic process sample that could represent surface roughness of a fabricated part. We consider both of these examples later. Other examples of the stochastic textured surface data include images of stone countertops (Liu and MacGregor 2006), ceramic capacitor surfaces (Lin 2007a), lumber surfaces (Bharati et al. 2003), and microscopy images of material microstructure samples (Torquato 2002, Liu and Shapiro 2015). Recall that point cloud surface roughness data of machined, cast, or formed metal parts is also an example of the stochastic textured surface data, but we illustrate the approach with image data throughout.

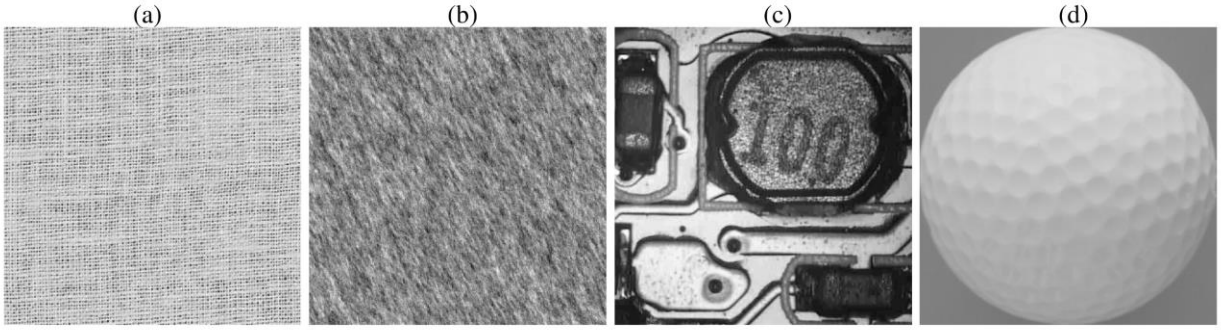


Figure 2.1. Images of (a) a textile fabric sample, (b) a simulated 2-D stochastic process, (c) components on a circuit board, and (d) a golf ball. The first two are examples of stochastic textured surfaces to which the approach of this chapter applies.

The stochastic textured surface data have a distinguishing characteristic that renders most of

the SPC literature for profile data largely inapplicable. Such profile SPC works include Woodall et al. (2004), Zou et al. (2007, 2008), Chicken et al. (2009), Jensen and Birch (2009), Chang and Yadama (2010), Qiu et al. (2010), Qiu and Zou (2010), Wells et al. (2012), Xu et al. (2012), Yu et al. (2012), Zou et al. (2012), Paynabar et al. (2013), Viveros-Aguilera et al. (2014), Grasso et al. (2016), and Paynabar et al. (2016). Profile monitoring works such as these are inapplicable to our stochastic textured surface problem, because they require images (i.e., 2-D profiles) for which there is some "gold standard" image comprised of distinct features that represent normal behavior of the process. Typically, the gold standard image would be closely related to the mean image, where the mean is across a sample of multiple images at the same location. For example, in detecting missing or miss-positioned components in a printed circuit assembly, the gold standard is an image of a complete assembly with all the components assembled correctly, as shown in Figure 2.1(c). In defect defection on smooth metallic surfaces, the gold standard is trivially a non-textured surface of a constant intensity. In monitoring stamping press tonnage signatures (Jin and Shi 1999), the gold standard is the ideal tonnage signature over the course of one stamping cycle that results when the process is behaving normally. For monitoring surfaces that have deterministically repeated patterns under ideal behavior, such as the dimpled surface of the golf ball in Figure 2.1(d), the gold standard is an image of a golf ball from computer aided design representations or from normal process behavior, perhaps after properly registering and aligning (for image registration techniques, see Xing and Qiu 2011; Qiu and Xing 2013a, 2013b).

In stark contrast, there is no such gold standard image for stochastic textured surfaces like those depicted in Figures 2.1(a) and (b), because the specific configuration of pixel greyscale values varies stochastically from image to image under normal process behavior. In other words, there are infinitely many stochastic textured surface images that have exactly the same normal stochastic behavior, but are all completely different images that do not match pixel-to-pixel. Moreover, they cannot be easily aligned, transformed or warped into a common gold standard image, because of

the stochastic nature of the surface. One might consider defining the gold standard image as the spatial mean function for the image (where the mean is taken across an ensemble of images of the same size). However, because of the stationary stochastic nature of the surfaces we consider, the spatial mean function for an image would have the same constant greyscale intensity value for every pixel in the entire image. In other words, the only possible gold standard image would have the same greyscale intensity for every pixel, and any comparison of the inspection images to this gold standard image would have little relevance for detecting defects.

Standard parametric random field models such as Gaussian random fields (Rasmussen and Williams 2006) lack the flexibility to capture the complex dynamics of many stochastic textured surfaces. For example, the warps and wefts of the textile in Figure 2.1(a) have components that resemble spatial periodicity, but their distances and thicknesses are much too random to be modeled as periodic, and there are additional random components on top of this. If the spacing between the warps and wefts were more deterministically repeatable (like the deterministic spacing of the golf ball dimples in Figure 2.1(d)), then this periodic component might be modeled as a legitimate profile mean function and handled via existing profile monitoring methods. Nonetheless, the random nature of the spacing precludes this approach.

Theoretically, the joint distribution of the collection of pixels in a stochastic textured surface sample provides a complete statistical representation, including capturing any stochastic spatial dynamics. Direct estimation of such a high-dimensional nonparametric distribution is obviously infeasible. However, under the stationary Markov random field assumptions of Section 2.2, the joint distribution can be implicitly and approximately characterized by the conditional distribution of individual element/pixel values in the image, given the values of a collection of neighboring pixels, with the conditional distribution estimated using supervised learning methods applied to an in-control training image sample(s). This technique was used in Bostanabad et al. (2016) to characterize and reconstruct binary material microstructure images. In this work, we use this

supervised learning approach to obtain an implicit representation of the in-control (i.e., normal) statistical behavior of a stochastic textured surface. Our objective is to develop a statistical monitoring approach for detecting local phenomena or defects in the manufactured stochastic textured surfaces that are statistically inconsistent with the in-control behavior, as represented by the implicit supervised learning characterization.

There is a growing body of work on image monitoring (see, e.g., the review paper of Megahed et al. 2011). Some early works directly monitored the intensity levels of the pixels in the images (Jiang and Jiang 1998, Armingol et al. 2003). Most of later methods first extracted a small set of predefined feature characteristics from the images and then monitored directly those specific characteristics or the statistics obtained from them. Common characteristics include length, width, and area (Tan et al. 1996), shape (Liang and Chiou 2008), and diameter (Lyu and Chen 2009) of specific features identified in the image. Other work has monitored frequency domain characteristics based on wavelets (Liu and MacGregor 2006, Lin 2007a, Lin 2007b) and frequency spectrum features (Tunák et al. 2009), principle components (Bharati and MacGregor 1998, Bharati et al. 2003), and grey level co-occurrence matrix features (Tunák and Linka 2008). Recently, Megahed et al. (2012) monitor summary statistics comprised of the average intensity levels of predefined windows of various sizes across the images.

Using predefined features is problem-specific, by definition, and requires that the users have a fairly specific idea of the nature of the defects that they would like to detect. Our goal is to develop a more general approach that can detect general local deviations from the normal in-control statistical behavior of stochastic textured surfaces, where the normal in-control statistical behavior is modeled in a reasonably generic manner from the in-control training images. In this regard, the approach is analogous to the classic Shewhart control charting approach (Montgomery 2009), which generically characterizes in-control behavior from a training sample (Phase I), and monitors future samples to detect general departures from the in-control behavior (Phase II). Our primary

monitoring statistic is derived from some appropriate spatial moving statistic (to be defined in Section 2.3), computed from the residual errors of the supervised learning model that characterizes the stochastic textured surface of interest. In addition to monitoring, our approach is designed to help users diagnose the cause of the deviations from normal behavior via highlighting pixels with large spatial moving statistics.

To the best of our knowledge, most industrial machine vision algorithms are intended for situations in which there is a legitimate gold-standard image and/or there are distinct predefined features (e.g., edges, corners, circles, spectral peak frequencies/amplitudes, average intensity levels, etc.) that can be detected with standard image processing toolboxes. The types of stochastic textured surfaces to which this work applies have neither a gold standard image nor standard features that can be detected. The main contribution of this work is developing an approach that can be used for this general class of stochastic textured surface inspection images, for which there is currently a hole in the existing literature.

It should be noted that, although our algorithm could be applied to monitoring surfaces with deterministically repeating patterns like the dimpled surface in Figure 2.1(d), we do not recommend it for that. A much more sensible approach would take into account the known, deterministic spacing and size of the dimples to either (i) compare inspection images to a gold standard dimpled surface representing the nominal geometry (after proper registration and alignment of the images) or (ii) extract relevant features related to the dimple size and spacing and monitor the features.

The remainder of the chapter is organized as follows. Section 2.2 describes how supervised learning can be used to implicitly characterize the normal in-control spatial statistical behavior of the stochastic textured surfaces. Section 2.3 introduces two spatial moving statistics and the primary monitoring statistic. Section 2.4 elaborates details of our monitoring and diagnostic algorithm. Sections 2.5 and 2.6 illustrate and compare the approach with three other approaches in

a simulation study and in the textile example depicted in Figure 2.1(a), respectively. Section 2.7 concludes this chapter.

2.2 Modeling the Spatial Statistical Characteristics of the Stochastic Textured Surfaces via Supervised Learning

Suppose an image is comprised of M pixels, and let $\mathbf{Y}_j = \left[y_{j,1},\ y_{j,2},\ \ldots, y_{j,M}\right]^T$ (j = 1, 2, . . ., N) denote the set of ordered pixels for the j^{th} image in a sample of N images. We use the subscript j later for indexing images; however, we will often omit it for simplicity, unless necessary. Suppose the elements of $\mathbf{Y}$ are ordered in a row raster scan pixel sequence of left-to-right, moving from the top row to the bottom row of the image, as illustrated in Figure 2.2 Let $f(\mathbf{Y})$ denote the joint distribution of $\mathbf{Y}$, which theoretically provides the most complete characterization of the statistical behavior of the stochastic textured surface. However, it is clearly infeasible to estimate such high-dimensional nonparametric distributions directly. In light of this, consider the factorization:

$$f(\mathbf{Y}) = f(y_M|y_{M-1}, y_{M-2}, \ldots)f(y_{M-1}|y_{M-2}, y_{M-3}, \ldots)\ldots f(y_2|y_1)f(y_1) = \prod_{i=1}^{M} f\left(y_i\middle|\mathbf{Y}^{(i)}\right),$$

where $\mathbf{Y}^{(i)} = \{y_k: k = 1,\ldots, i-1\}$. The notation is illustrated in Figure 2.2.

Using this factorization, we can implicitly obtain the joint distribution $f(\mathbf{Y})$ by learning each conditional distribution $f(y_i|\mathbf{Y}^{(i)})$ via fitting some appropriate supervised learning model to predict the "response" variable y_i as a function of the set of "predictor" variables $\mathbf{Y}^{(i)}$. Without further assumptions, this is unmanageable, in part because it would require learning M separate models, and many of them have an extremely high-dimensional predictor space (e.g., $\mathbf{Y}^{(M)}$ is $(M-1)$-dimensional). To make the problem more manageable, we assume the following Markov random field (MRF) properties for the stochastic textured surfaces, which are generally quite reasonable and are often assumed in texture synthesis problems (Efros and Leung 1999, Levina and Bickel 2006). The first MRF property is *locality:* there exists a neighborhood $\mathbf{y}^{(i)} = \{y_k \in \mathbf{Y}^{(i)}$: pixel k is within some neighborhood of pixel $i\}$ such that given $\mathbf{y}^{(i)}$, y_i is independent of all other pixels in

$\mathbf{Y}^{(i)}\backslash\mathbf{y}^{(i)}$, i.e., such that $f(y_i \mid \mathbf{Y}^{(i)}) = f(y_i \mid \mathbf{y}^{(i)})$. Figure 2.2 depicts this neighborhood $\mathbf{y}^{(i)}$ as the shaded region. The second MRF property is s*tationarity*: $f(y_i = y \mid \mathbf{y}^{(i)} = \mathbf{y})$, as a function of y and $\mathbf{y}$, is independent of pixel location i.

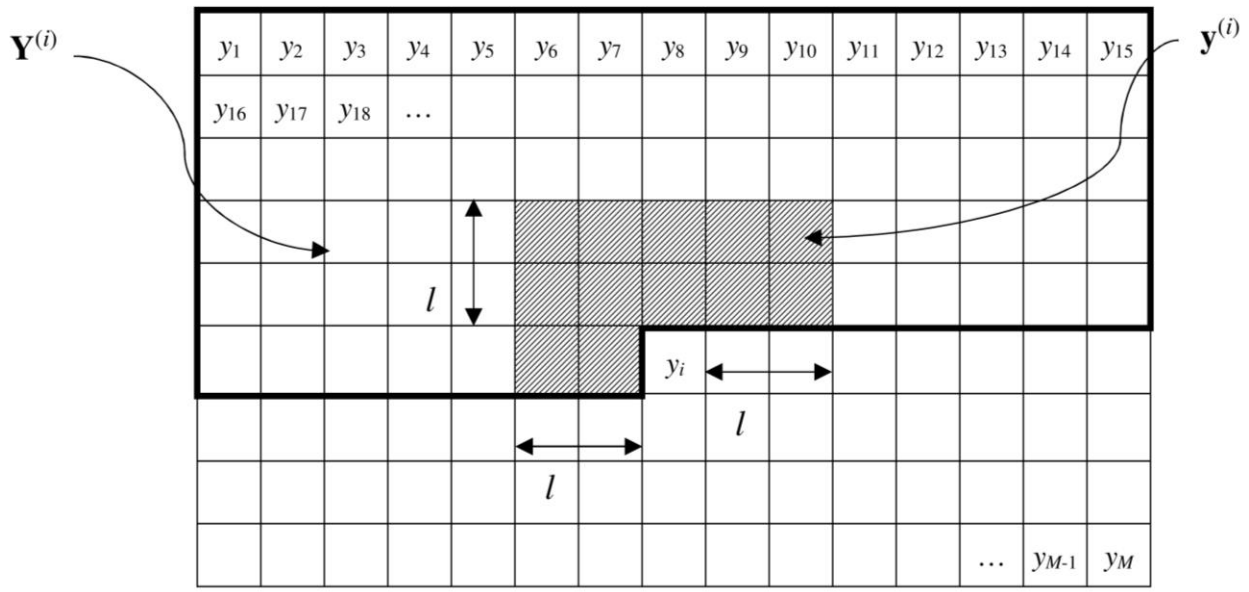


Figure 2.2. Illustration of the notation with a stylized pixelated image (each cell represents a pixel). The pixels inside the area with bold borderlines are the elements of $\mathbf{Y}^{(i)}$, and the shaded pixels are the elements of $\mathbf{y}^{(i)}$. Given $\mathbf{y}^{(i)}$, y_i is assumed independent of $\mathbf{Y}^{(i)}\backslash\mathbf{y}^{(i)}$.

By the locality assumption,

$$f(\mathbf{Y}) = \prod_{i=1}^{M} f\left(y_i \middle| \mathbf{Y}^{(i)}\right) \approx \prod_{i=1}^{M} f\left(y_i \middle| \mathbf{y}^{(i)}\right). \tag{2.1}$$

Thus, we can obtain $f(\mathbf{Y})$ by learning $f(y_i|\mathbf{y}^{(i)})$, which is more computationally feasible since the size of $\mathbf{y}^{(i)}$ is much smaller than that of $\mathbf{Y}^{(i)}$. The stationarity assumption enables us to estimate $f(y_i|\mathbf{y}^{(i)})$ by fitting an appropriate supervised learning model to a set of training data constructed from the collection of pixels in some training image $\mathbf{Y}$. The training data array consists of M rows with each row corresponding to one of the pixels in $\mathbf{Y}$. The i^{th} row of the training data set is comprised of $\{y_i, \mathbf{y}^{(i)}\}$, where y_i and $\mathbf{y}^{(i)}$ represent the response and predictor variables, respectively.

When fitting the supervised learning model, the first column is treated as the response column, and the remaining columns as the predictor columns. Henceforth, M denotes the number of pixels in the interior of the image, excluding a small boundary region just large enough that the first pixel y_1 has a full-size neighborhood $\mathbf{y}^{(1)}$.

If the greyscale pixel values were coarsely discretized, then the conditional distribution of $y_i \mid \mathbf{y}^{(i)}$ would be multinomial, and any appropriate supervised learning classifier could be used to learn the multinomial probabilities as a function of the predictor variables $\mathbf{y}^{(i)}$. For the case of binary images representing two-phase material microstructure samples, Bostanabad et al. (2016) used this supervised learning approach to learn the Bernoulli conditional probabilities of $y_i \mid \mathbf{y}^{(i)}$. Their fitted supervised learning model provided an implicit characterization (via (2.1)) of the microstructure, which they used to reconstruct microstructure samples that were statistically equivalent to the original training sample.

Because we are assuming finely discretized greyscale intensity levels, we treat them as continuous and consider a supervised learning model of the general form

$$y_i = g\left(\mathbf{y}^{(i)}\right) + \varepsilon_i, \tag{2.2}$$

where $g\left(\mathbf{y}^{(i)}\right)$ is the mean of the conditional distribution $f(y_i|\mathbf{y}^{(i)})$, and ε_i is a zero-mean error. Applying an off-the-shelf supervised learning algorithm to an in-control image, we obtain a model that represents the estimated conditional mean function $\hat{g}\left(\mathbf{y}^{(i)}\right)$. Although the conditional mean does not fully represent the conditional distribution, it does provide rich enough information to monitor for deviations from the in-control statistical behavior of the stochastic textured surfaces. As will be discussed in Section 2.3, we use the residuals of the supervised learning model for our monitoring and diagnostic purposes.

It should be noted that other ways of ordering the pixels, such as the zigzag scanning method used in Megahed and Camelio (2012), could result in a different fitted supervised learning model, especially if the surface is not isotropic. If desired, one could use cross-validation (CV) to select

the best ordering as the one that minimizes the CV error sum of squares. In all of our examples, we only considered the raster scan order depicted in Figure 2.2.

2.3 Choice of Monitoring Statistic

In this section, we develop our monitoring statistics that are based on the residuals of the in-control supervised learning model. Section 2.3.1 presents the monitoring approach in terms of a general spatial moving statistic (SMS) that appropriately aggregates the local residual behavior, and Sections 2.3.2 and 2.3.3 discusses two specific statistics to serve as the SMS.

2.3.1 Monitoring based on local residual behaviors

Henceforth, let $g\left(\mathbf{y}^{(i)}\right)$ denote the conditional mean model fitted to a training image(s) that are known to represent in-control behavior. For a new inspection image, denote the residual for the i^{th} pixel (i = 1, 2, . . ., M) by

$$r_i = y_i - \hat{g}(\mathbf{y}^{(i)}). \tag{2.3}$$

Note that the residuals themselves constitute an image that corresponds pixel-wise to the image from which the residuals are computed (e.g., see Figure 2.3). If the new image also behaves as under the in-control conditions, then the residuals $\{r_i: i = 1, 2, \ldots, M\}$ should behave approximately as white noise, although departures from white noise are automatically adjusted for, via the way the control limits are determined in our approach (see Section 2.4.1). In contrast, if the new image has defects or other departures from the in-control stochastic behavior, then the residuals should behave differently than that when the image is in-control. Hence, our monitoring procedure is based on monitoring the residuals in a manner to be described shortly (see the online supplement of Bui and Apley (2018a) for further discussion on the types of defects that our algorithm can detect, which are reasonably general).

Monitoring individual residuals may not be sensitive enough to detect milder defects, for the same reason that Shewhart individual charts are not sensitive enough to small mean shifts.

Consequently, we use moving window to aggregate the residuals in some manner over an appropriately sized spatial neighborhood of the image. To measure the degree of deviation from in-control behavior of the residuals within the moving window neighborhood, we use a SMS that is some statistic computed from the residuals within a spatial moving window that is scanned across the residual image. Let w denote the width (in number of pixels) of the square spatial moving window, which contains $n = w^2$ residuals. For example, the window with the bold border in Figure 2.3 depicts the moving window centered at the i^{th} pixel. The SMS at the i^{th} pixel of the j^{th} image, denoted by $SMS_{j,i}$, is defined as some function of the w^2 residuals within the moving window surrounding the i^{th} pixel of the j^{th} image. In this chapter, we consider two such SMSs that are intuitively appealing and that we have found to result in good defect detection performance in our examples: (i) a one-sample Anderson–Darling (A-D) statistic and (ii) a Box–Pierce (B-P) type statistic, which we describe in Sections 2.3.2 and 2.3.3, respectively.

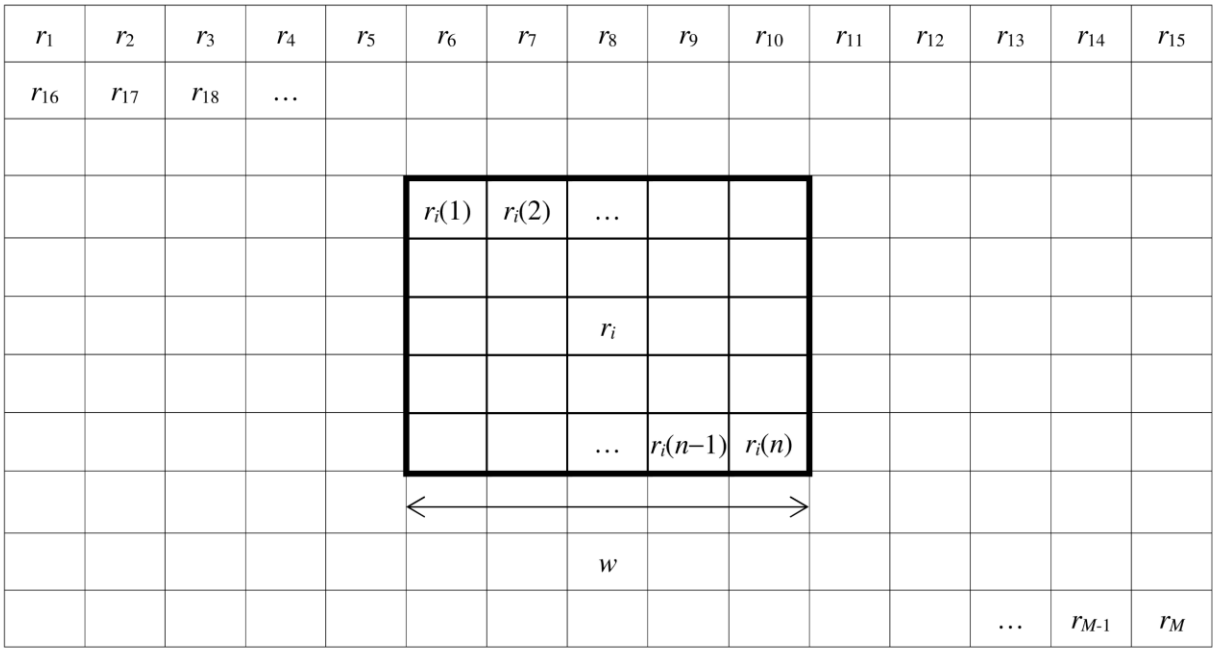


Figure 2.3. An image of residuals illustrating the spatial moving windows: each cell corresponds to a pixel of the image from which the residuals are computed. The pixels $\{r_i(1), r_i(2), \ldots, r_i(n)\}$ inside the bold lines are the elements of the square moving window of $n = w^2$ residuals centered at the i^{th} pixel, corresponding to residual $r_i \equiv r_i((w^2 + 1)/2)$.

Our algorithm is intended for monitoring and diagnosing individual images using a single aggregate summary statistic for each image. Moreover, the intent is that an alarm will be sounded if an individual image contains a defect, as opposed to requiring that defects persistently occur across a consecutive set of images. In this respect, our approach is akin to a Shewhart individual chart. We define our monitoring statistic for the j^{th} image to be

$$S_j = \max_{i=1\ldots M} SMS_{j,i} \tag{2.4}$$

If the defects do occur persistently across consecutive images, then our monitoring approach could be enhanced by using an EWMA-type or CUSUM-type accumulation of S_j, although we do not pursue this in this chapter.

2.3.2 A-D Statistic

As discussed in Section 2.3.1, we expect a local change in the distribution of the residuals in the defect region, relative to the in-control residual distribution. We represent the latter by a reference cumulative density function (cdf), denoted by φ, of all the residuals **R** computed from a representative in-control image(s). As a statistic that measures the deviation (from φ) of the residual distribution within some neighborhood of a pixel, we consider a one-sample A-D statistic (Anderson and Darling 1954), which compares the empirical cdf of the residuals within a spatial moving window versus φ. We also considered a one sample Kolmogorov-Smirnov statistic, but do not pursue it here, because we found that the A-D statistic performed better. This perhaps was because the A-D statistic is more sensitive to changes in the tails of the distribution, which correspond to large-magnitude residuals.

Let the residuals $\{r_i(1), r_i(2), \ldots, r_i(n)\}$ within the moving window around the i^{th} pixel be ordered from smallest to largest. The one-sample A-D SMS at the i^{th} pixel is defined as:

$$A_i^2 = -n - \sum_{k=1}^{n} \frac{2k-1}{n} \ln\{\varphi(r_i(k))[1 - \varphi(r_i(n+1-k))]\} \tag{2.5}$$

Since the sample size for the training image is quite large, one might consider using the

empirical cdf of the residuals **R** (denoted by *F*, for which a corresponding histogram is shown in Figure 2.4) for the training image as φ in (2.5). However, this causes a potential problem, because the one-sample A-D statistic is infinite/undefined if any of the *n* elements within the moving window are beyond the support of φ. To illustrate, Figure 2.4 shows a histogram of approximately 0.25 million residuals from a training image in one of our examples, the support of which extends from [–2.45, 2.84]. Thus, if a new image contains a residual that falls outside the interval [–2.45, 2.84], which happened frequently in our example (even with the process was in-control), the statistic in (2.5) is infinite for any moving window containing that residual.

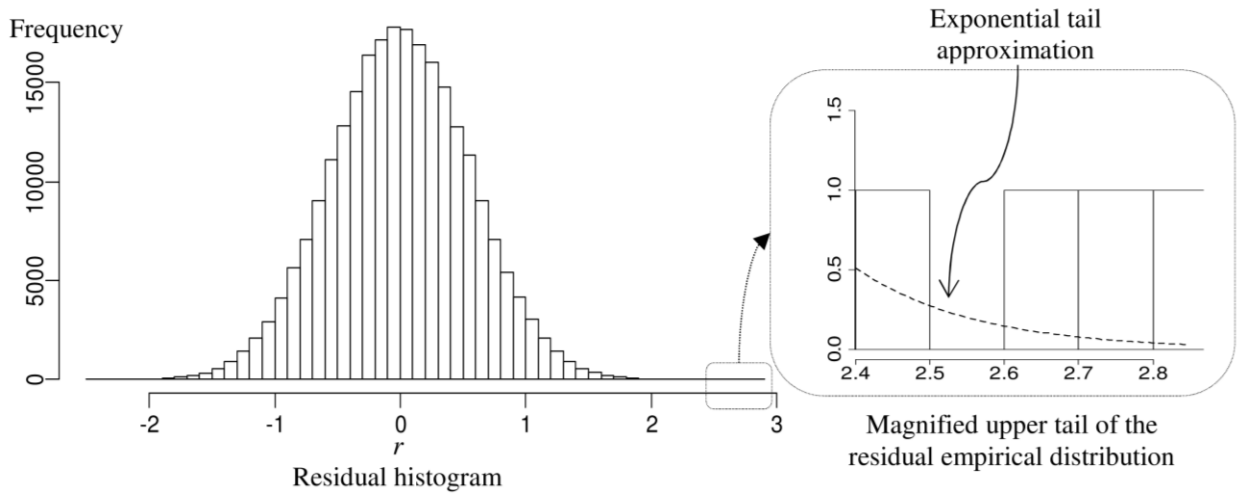


Figure 2.4. Approximating the upper and lower tails of the residual empirical cdf with an exponentially decaying distribution. The sample size is approximately 0.25 million pixels.

To avoid this problem, instead of using $\varphi = F$ directly, we replace its upper and lower tails with an exponentially decaying tail approximation. The upper tail approximation for $r > 2.38$ is illustrated in Figure 2.4. More specifically, let r_{q_l} and r_{1-q_u} denote the lower q_l quantile and upper q_u quantile of F for some small probabilities q_l and q_u such that $F(r_{q_l}) = q_l$ and $F(r_{1-q_u}) = 1 - q_u$. The probabilities q_l and q_u should be large enough to have enough tail observations to get a good estimate of the exponential rate parameters for the tail approximation, but otherwise as small as possible. For our examples we have used values $q_l \approx q_u \approx 1.6 \times 10^{-3}$, which, because of the

large number of pixels in typical images, translate to around 400 observations in each tail. To estimate the rate parameters, we fit the observations corresponding to the lower and upper tails of F with the exponential probability density functions (pdfs):

$$f(r) = \begin{cases} \frac{q_l}{\lambda_l} exp\left\{\frac{r-r_{q_l}}{\lambda_l}\right\} & : \quad r \le r_{q_l} \\ \frac{q_u}{\lambda_u} exp\left\{-\frac{r-r_{1-q_u}}{\lambda_u}\right\} & : \quad r \ge r_{1-q_u} \end{cases}$$

The maximum likelihood estimators of the lower and upper exponential rate parameters are $\lambda_l = r_{q_l} - ave\{r_i:\ r_i \le r_{q_l}\}$, and $\lambda_u = ave\{r_i:\ r_i \ge r_{1-q_u}\} - r_{1-q_u}$. We then choose a very small probability p ($p = 5/M$ in the example in Figure 2.4) and replace $F(r)$ by its exponential tail approximation for $r < r_p$ and $r > r_{1-p}$, where r_p and r_{1-p} are the lower and upper p quantiles of F. That is, for φ in (2.5) we use

$$\varphi(r) = \begin{cases} p \times exp\left\{\frac{r-r_p}{\lambda_l}\right\} : & r \le r_p \\ F(r) : & r_p < r < r_{1-p} \\ 1 - p \times exp\left\{-\frac{r-r_{1-p}}{\lambda_u}\right\} : & r \ge r_{1-p} \end{cases}$$

2.3.3 B-P Type Statistic

A B-P (aka portmanteau) test (Box and Pierce 1970) is widely used for testing the existence of autocorrelations in time series. Likewise, a B-P type statistic can be used to detect spatial correlations in our stochastic textured surface images. Because local defects in the stochastic textured surfaces are likely to result in local spatial correlations in the residuals, the B-P type statistic is intuitively appealing for our objective. We define the B-P type SMS for the i^{th} pixel as

$$T_i = \sum_{k=1}^{n} \widehat{Cov}_{i,k}^2 \tag{2.6}$$

where $\widehat{Cov}_{i,k}^2$ is some local estimate of the covariance between the residual r_i at the i^{th} pixel and another residual r_k within the moving window of n pixels surrounding the i^{th} pixel (e.g., the moving window in Figure 2.3). Note that $\widehat{Cov}_{i,i}^2$ is included in T_i in (2.6). To estimate $\widehat{Cov}_{i,k}$, we use a

kernel weighted window centered at the i^{th} pixel. For ease of illustration, let i_1 and i_2 be the row and column indices of the i^{th} pixel, and let k_1 and k_2 be the row and column indices of the k^{th} pixel. Then,

$$\widehat{Cov}_{i,k} = \frac{\sum_{h=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} K(h,m) r_{i_1-h,i_2-m} r_{k_1-h,k_2-m}}{\sum_{h=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} K(h,m)}$$

where $K(h,m)$ is the Epanechnikov quadratic kernel:

$$K(h,m) = \begin{cases} \frac{3}{4}\left(1 - \frac{h^2+m^2}{\left(\frac{w+1}{2}\right)^2}\right) : \; h^2 + m^2 \leq \left(\frac{w+1}{2}\right)^2 \\ 0 \qquad\qquad : \quad otherwise \end{cases} \tag{2.7}$$

2.4 Stochastic Textured Surface Monitoring and Diagnostic Algorithm

Our approach involves two stages: monitoring and diagnosis. The monitoring stage has two phases. The first is an offline training phase (Phase I) that constructs a control limit based on the empirical distribution of the monitoring statistic computed for a set of in-control images. As defined in (2.4), the monitoring statistic for the j^{th} image is the maximum of the SMS values across all pixels in that image, where our SMS for pixel i in image j is either A_i^2 or T_i for image j. The SMS values are calculated from the residuals of the supervised learning model that characterizes the stochastic textured surface to be monitored. The second phase is an online monitoring phase (Phase II) that computes a monitoring statistic for each new image similarly to that in Phase I. If the monitoring statistic is beyond a control limit, an alarm is sounded, and the diagnostic stage is invoked. The diagnostic stage constructs a binary image that corresponds pixel-to-pixel with the original image and highlights the pixels with SMS values larger than some threshold. We refer to these as *diagnostic images*. We provide details of Phase I of the monitoring stage in Section 2.4.1. Phase II of the monitoring stage, as well as the diagnostic stage, are discussed in Section 2.4.2.

2.4.1 Phase I of the Monitoring Stage: Fitting the Supervised Learning Model and Establishing the Control Limits

Phase I of the monitoring stage begins with choosing a region from an image (or images) that is known to be in-control, from which to construct the training data set for fitting the supervised learning model as described in Section 2.2. The neighborhood structure must also be chosen. This can be flexible, but to simplify the discussion, we define the neighborhood by a single parameter l, which is the number of pixels to the right/left and above the response pixel, corresponding to our raster scan method. Figure 2.2 illustrates such a neighborhood with $l = 2$. The value of l should be large enough to include all important predictor variables (such that the MRF locality property holds), but not so large as to incur unnecessary computational expense. We recommend choosing l via CV during the process of fitting the supervised learning model to minimize some measure of CV error. Readers are referred to Bostanabad et al. (2016) for further details of the model fitting procedure.

After fitting the supervised learning model, we apply it to a new image j (that is believed to be in-control) to calculate the predictions $\left\{\hat{g}\left(\mathbf{y}_j^{(i)}\right): i = 1, 2, \ldots\right\}$ and the corresponding residual errors $\left\{r_{j,i}: i = 1, 2, \ldots\right\}$ via (2.3). After that, the SMS values $\left\{SMS_{j,i}: i = 1, 2, \ldots\right\}$ are calculated as described in Section 2.3. Finally, the monitoring statistic S_j for the j^{th} image is computed via (2.4). This process is repeated for a set of N in-control images (i.e., for $j = 1, 2, \ldots, N$) to give a sample $\{S_j: j = 1, 2, \ldots, N\}$ of monitoring statistics that represent the in-control state. Given the complexity of the supervised learning model and the image texture characteristics, it is not possible to derive some exact (or even reasonably approximate) theoretical distribution of the residuals and the resulting theoretical distribution of the monitoring statistic S in order to set the control limits. Thus, we set the control limits based on the empirical distribution of $\{S_j: j = 1, 2, \ldots, N\}$ from the set of in-control Phase I images, which is often available in practice.

Figure 2.5 shows histograms of $\{S_j: j = 1, 2, \ldots, N\}$ for the $N = 1{,}000$ Phase I images for the example in Section 2.5 based on A-D (Figure 2.5(a)) and B-P type (Figure 2.5(b)) SMSs, respectively. The theoretical support of the distribution of S in either case is $[0, \infty)$, and a larger S_j

indicates a higher likelihood that image j contains a defect. Thus, there is only an upper control limit. Letting α (e.g. $\alpha = 0.003$) denote the desired Type I error for an individual j^{th} image, we set the control limit as the $(1 - \alpha)$ quantile of the empirical distribution of $\{S_j: j = 1, 2, \ldots, N\}$.

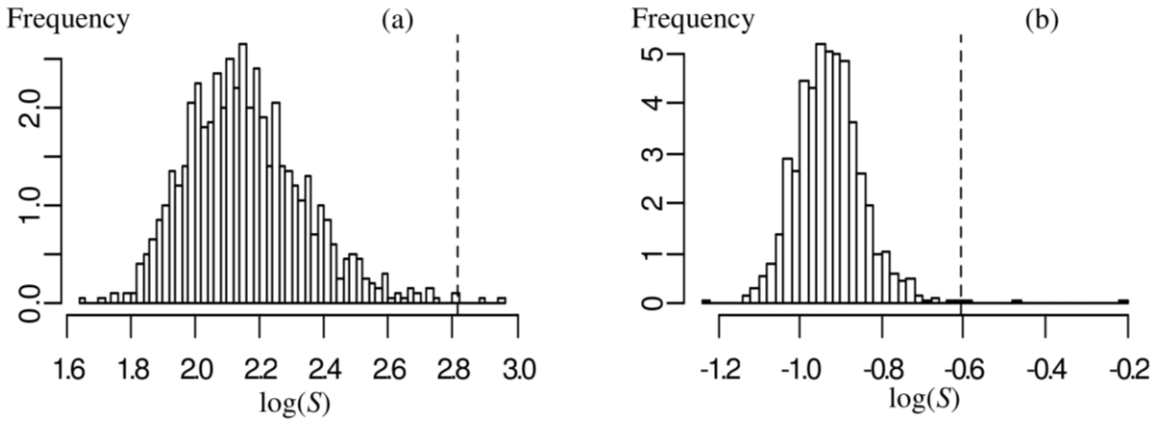


Figure 2.5. Histograms of Phase I $\{S_j: j = 1, 2, \ldots, N\}$ in log scale based on (a) the A-D statistic and (b) the B-P type statistic, computed from N = 1,000 in-control images for the example in Section 2.5 with w = 25. The dashed lines are the control limits corresponding to $\alpha = 0.003$.

2.4.2 Phase II of the Monitoring Stage and the Diagnostic Stage

Phase II of the monitoring stage, which involves many of the same calculations as Phase I, and the diagnostic stage are relatively straightforward. First, a data array is constructed from each new image as described in Section 2.2, and the supervised learning model fitted in Phase I is applied to generate the predictions $\hat{g}(\mathbf{y}^{(i)})$ and the corresponding residuals for the new image. The SMS values at each pixel of the new image are computed from these residuals, and the monitoring statistic S is computed via (2.4), after which it is compared to the control limit calculated in Phase I.

If an alarm is sounded for an image, the algorithm invokes the diagnostic stage, which compares the SMS values at all pixels in that image with a diagnostic threshold. A binary diagnostic image is then constructed by plotting every pixel with SMS value larger than the diagnostic threshold as a black pixel, and the remaining pixels in the image as white pixels. The diagnostic threshold has no connection to the control limit, as the former applies to the SMS values,

whereas the latter applies to the *S* values. Moreover, while the control limit is chosen to control the Type I error, the diagnostic threshold is chosen purely to facilitate visualization of the nature of the defect. Our recommended strategy for selecting the diagnostic threshold is so that it results in a small but acceptable level of noise (i.e., black pixels that are not associated with actual defects) in the diagnostic image. We accomplish this by setting the diagnostic threshold at the $(1 - n_D/M_{SMS})$ quantile of the empirical distribution of all SMS values computed for all Phase I images, where M_{SMS} is the number of SMS values computed for each Phase I image, and n_D is the desired average number of black (noise) pixels in a diagnostic image of an in-control image. The choice of n_D depends on the image size in general. We have used n_D ~ 5—10 for an image size of 250×250 pixels in our examples. Alternatively, instead of selecting a single n_D, users could vary n_D as the diagnostic image is dynamically changed to better facilitate visualization of the defect.

2.5 Simulation Study

In this section, we demonstrate and evaluate our approach with simulated images of the 2-D stochastic process depicted in Figure 2.1(b). We also compare its performance with three alternative methods. The images were generated via the spatial autoregressive model $y(i,k) = \phi_1 y(i-1,k) + \phi_2 y(i,k-1) + \varepsilon(i,k)$, where $y(i, k)$ denotes the image greyscale level at pixel location (i, k) with i and k the row and column indices, respectively. We used $\phi_1 = 0.6$, $\phi_2 = 0.35$, and ε a zero-mean Gaussian white noise. After generating the process, we translated/rescaled it to the interval [0, 255] to obtain the corresponding greyscale image for plotting purposes. For applying our algorithm, all images were subsequently standardized by subtracting from each pixel the average greyscale value for all pixels in that image and then dividing by the greyscale standard deviation for all pixels in the image. For real examples, this is helpful if the lighting conditions vary from image to image, although ideally the lighting should be controlled. Note that the MRF assumptions hold for the images in this example, by construction.

2.5.1 Monitoring Stage

We evaluated the monitoring performance of our algorithm with 10 replicates of the following experiment. On each replicate we first generated an image of size 500×500 pixels, similar to the one in Figure 2.1(b), for model fitting (discussed in Section 2.4.1). Then, we used a regression tree as the supervised learning model (any appropriate supervised learner could be used) because it is more computationally reasonable to fit for large training data sets. The neighborhood size l was obtained during the tree fitting process as the one that minimized the CV sum-of-squares error. This resulted in $l = 1$, which agrees with the lag-one autoregressive model used to generate the data. For real examples, like the textile application in Section 2.6, the CV procedure will typically select a much larger value of l.

To construct the control limit, we generated a Phase I set of $N = 1000$ in-control images, each of size 250×250 pixels, in the same manner as the training image used for model fitting. Using the fitted regression tree from the training image, for each image j in the Phase I set, we computed the SMS values for every pixel and then the monitoring statistic S_j in (2.4), as described in Section 2.4.1. We considered both the A-D and B-P type SMSs, each with several spatial moving window sizes ($w = 5$, 15, and 25), for comparison purposes. For the A-D statistic, we chose $q_l \approx q_u \approx 1.6 \times 10^{-3}$ in order to give around 400 observations in each tail for estimating the exponential tail parameters. We also chose $p \approx 2 \times 10^{-5}$, for which $r_p = -2.16$ and $r_{1-p} = 2.38$, and we replaced $F(r)$ by its exponential tail approximation for $r \notin (r_p, r_{1-p})$. From the empirical distribution of $\{S_j: j = 1, 2, \ldots, 1000\}$ (the histograms for which are shown in Figure 2.5 for $w = 25$) and with a desired Type I error rate of $\alpha = 0.003$, we selected the control limit, which depended on the choice of w.

For monitoring performance evaluation, we generated 400 Phase II images, all containing defects, by first generating an in-control image of size 250×250 pixels from the same spatial autoregressive model and then creating a defect and superimposing on the image. We considered "white noise defects" that were Gaussian white noise process (i.e., the spatial autoregressive

process with $\phi_1 = \phi_2 = 0$) with the same mean and standard deviation as the white noise ε in the in-control process. The defect regions that we superimposed were ellipsoidal shaped and of sizes 5×5, 5×21, 9×21, and 15×21 (the sizes refer to the lengths of the major and minor axes of the ellipses, which were aligned with the horizontal and vertical axes of the images). Randomly positioned defects of each these sizes were added to 100 images each (one defect added to each image) to generate the Phase II out-of-control images. The first row of Figure 2.6 shows some examples of these Phase II images, the defects of which are difficult to spot visually.

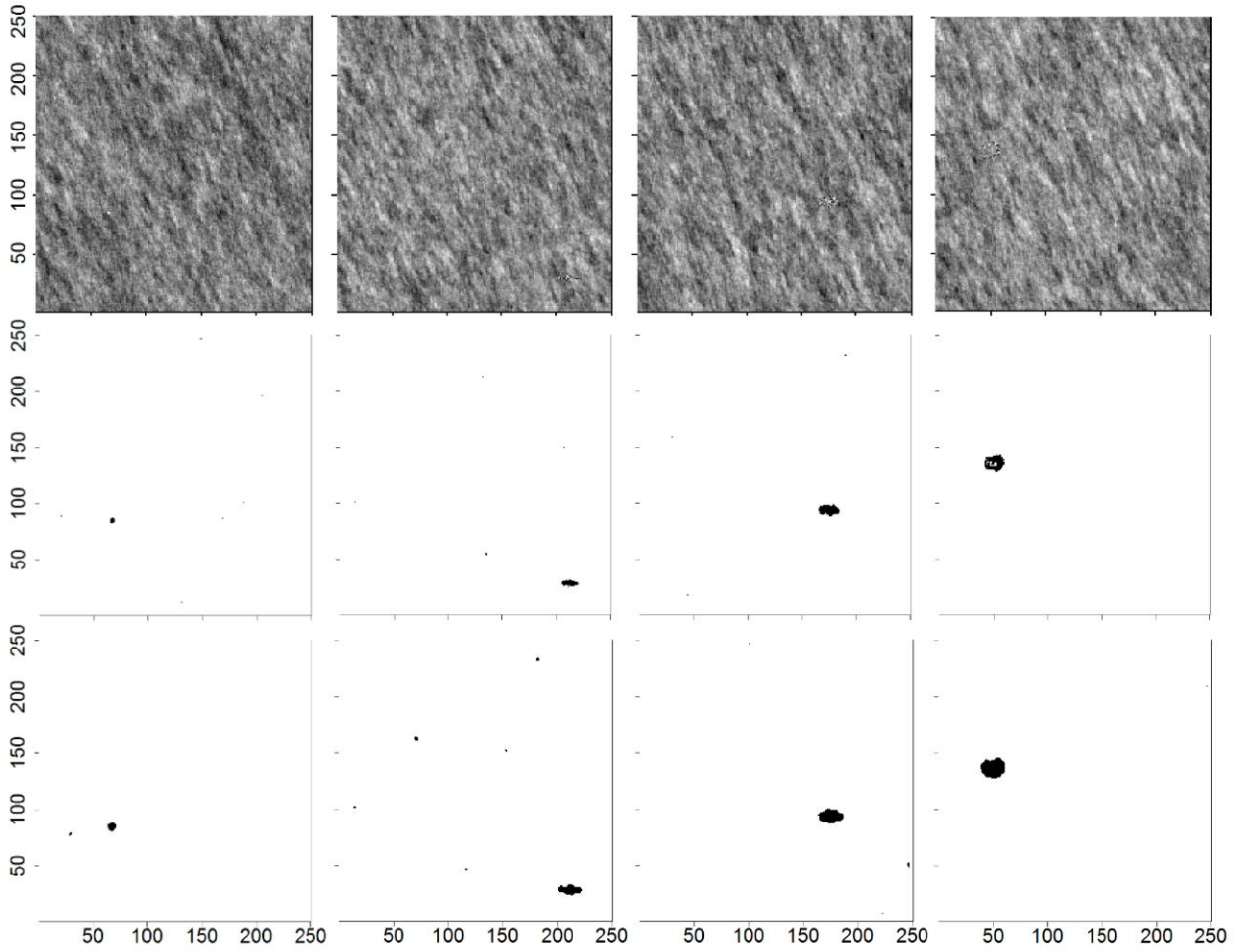


Figure 2.6. Phase II images in the simulation example containing white noise defects (top row) and their diagnostic images using the A-D-based statistic (middle row) and the B-P-type statistic (bottom row). The defects in the images in the top panels have different sizes: (1st column) 5×5, (2nd column) 5×21, (3rd column) 9×21, and (4th column) 15×21.

Table 2.1 reports the average power across 10 replicates for our approach (with different combinations of the SMS statistic and *w*) for all four defect sizes mentioned above. Notice that the algorithm generally detects defects with higher power when *w* is approximately the same size as the defects (we have also observed this phenomenon in other examples). This is intuitively reasonable, because the SMS (either A-D or B-P type) is larger when its window contains more pixels in the defect region and fewer pixels in the normal region. However, this is not an implementable guideline for choosing *w*, because defect sizes may not be known in advance. Regarding choice of *w*, we have observed that for the A-D-based statistic, the performance suffers more when *w* is larger than the defects than it does when *w* is smaller than the defects. This can be observed by comparing the three columns for the A-D-based statistic in Table 2.1. Consequently, for the A-D-based statistic, we recommend choosing *w* to be approximately the same as the smallest defect size of interest.

Table 2.1. Average powers in 10 replicates of our approach at $\alpha = 0.003$

Defect sizes	A-D			B-P		
	w = 5	*w* = 15	*w* = 25	*w* = 5	*w* = 15	*w* = 25
5×5	0.205	0.004	0.003	0.955	0.884	0.858
5×21	0.785	0.791	0.247	0.997	1.000	1.000
9×21	0.964	1.000	0.987	1.000	1.000	1.000
15×21	0.990	1.000	1.000	1.000	1.000	1.000

For the B-P-type statistic, the monitoring performance for all but the smallest defects was almost perfect even when *w* is larger than the defect size, as can be seen from the three columns for the B-P-type statistic in Table 2.1. To demonstrate the extent to which the monitoring statistics in these cases exceed the control limits, Figure 2.7(a) shows boxplots of $(\bar{S}_w - CL_w)/(UCL_w - CL_w)$ across the 10 replicates, where $\bar{S}_w$ is the average B-P-type monitoring statistic using an SMS size of *w* for all Phase II images containing defects of size 5×21. Figures 2.6(b) and 7(c) show similar boxplots, but for defect sizes 9×21 and 15×21, respectively. By comparing the boxplots in each panel of Figure 2.7 we see that as *w* increases, the monitoring

statistic tends to exceed the control limit by larger amounts, i.e., the monitoring performance of the B-P-type statistic improves with larger *w*.

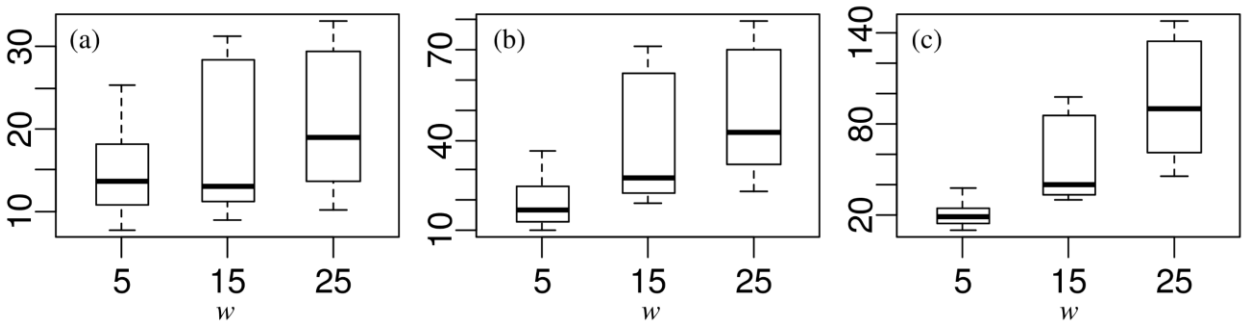


Figure 2.7. Boxplots of $(\bar{S} - CL)/(UCL - CL)$ across 10 replicates, where $\bar{S}$ is the average B-P-type monitoring statistic for all Phase II images containing defects of sizes: (a) 5×21, (b) 9×21, and (c) 15×21. Three window sizes *w* = 5, 15, and 25 were considered.

However, using a larger *w* for the B-P-type statistic has two potential drawbacks. First, a larger *w* requires more computational expense, because the number of covariance terms to be computed in each moving window increases quadratically in *w*, and the kernel window is also larger. Second, and perhaps more seriously, using larger *w* means that more boundary pixels (≈ *w*/2 pixels at each image edge) cannot be monitored, because full windows are required for computing the SMS. This is the reason why the monitoring performance of the B-P-based approach in Table 2.1 mildly degrades as *w* increases for the smallest defects, which are more likely to occur at the boundary as *w* increases. Therefore, for the B-P-based approach, we recommend that users choose *w* as large as possible while balancing the above drawbacks.

We also compared our algorithm with three alternative methods. The first uses the Epanechnikov quadratic kernel in (2.7) to compute the weighted average of pixel intensities within a moving window, and this is used as the SMS. Specifically, the SMS of the i^{th} pixel in an image is:

$$SMS_i = \frac{\sum_{h=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} K(h,m) y_{i_1-h,i_2-m}}{\sum_{h=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} K(h,m)},$$

where, as in Section 2.3.3, i_1 and i_2 are the row and column indices of the i^{th} pixel. Similar to our approach, the monitoring statistic for each image for this approach is the maximum SMS_i in (2.4) over all pixels in the image. We refer to this as the EPWMA approach. The second method, which we refer to as the EPWMV approach, is the same except that the SMS statistic is.

$$SMS_i = \frac{\sum_{h=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} K(h,m)\left(y_{i_1-h,i_2-m}-\bar{y}_i\right)^2}{\sum_{h=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} K(h,m)},$$

where $\bar{y}_i$ is unweighted mean of all pixel intensities in the window centered at pixel i.

The third method is the Haar-wavelet-based algorithm of Lin (2007a). Their algorithm divides a given image into many subimages and computes a monitoring statistic for each subimage, based on the 2-D Haar wavelet transform applied to these subimages. This is an example of the predefined-feature-based algorithms, where the features are defined by the 2-D Haar wavelet characteristics obtained from the subimages. The Lin (2007a) algorithm is not a standard control charting algorithm as defined in Megahed et al. (2011), because Lin (2007a) applies a spatial control chart within each image, as opposed to having a single charting statistic associated with each image. Thus, to have a common basis for comparison, we modify the Lin (2007a) approach as follows. The charted statistic for each image is taken to be the maximum of all of the Lin (2007a) statistics computed for all subimages of the image.

Analogous to Table 2.1, Table 2.2 reports the average power at a Type I error rate of $\alpha = 0.003$ (the same α used for our approach) across 10 replicates, but for the EPWMA and EPWMV approaches (with w = 5, 15, and 25) and the modified version of Lin (2007a). For each method, the control limits are chosen to control the Type I error based on a set of in-control images, analogous to how the control limit is selected for our algorithm. None of these approaches successfully detects the defects in this example.

2.5.2 Diagnostic Stage

If the monitoring stage signals an alarm, the algorithm invokes the diagnostic stage, which uses

the SMS values computed from Phase II of the monitoring stage and compares them with the diagnostic threshold(s) as discussed in Section 2.4.2. For illustration, in the second and third rows of Figure 2.6, we plotted the diagnostic images for the four defect-containing images in the top row of Figure 2.6, using A-D-based and B-P-type statistics, respectively, both with $w = 5$ (see the online supplement of Bui and Apley (2018a) for analogous results for $w = 15$ and 25). To set the diagnostic threshold, we used $n_D = 10$ (equivalent to having an average of 10 noise-related black pixels in the in-control diagnostic images) and the empirical distribution of the SMS statistics computed for all pixels in all 1000 Phase I images.

Table 2.2. Average power across 10 replicates for the EPWMA, EPWMV, and Lin (2007a) methods for the same example depicted in Table 2.1.

Defect	EPWMA			EPWMV			Lin
sizes	$w = 5$	$w = 15$	$w = 25$	$w = 5$	$w = 15$	$w = 25$	(2007a)
5×5	0.005	0.004	0.006	0.137	0.006	0.007	0.005
5×21	0.004	0.005	0.002	0.123	0.005	0.003	0.006
9×21	0.007	0.004	0.003	0.110	0.002	0.004	0.007
15×21	0.005	0.005	0.006	0.078	0.008	0.002	0.012

2.6 Textile Application

Next, we apply our approach to a set of real image data for a textile material, an example image of which is shown in Figure 2.1(a). Note that the fabric pattern is quite complex (as a stochastic process) with random thicknesses of and distances between fiber strands. Figure 2.8 displays six images containing defects that were created by physically scuffing, tearing, or otherwise deforming the fibers locally to represent a variety of defect types. All images were also standardized as a preprocessing step in this example. All the image data in this example are available in the R data package **textile** (Bui and Apley 2017a).

Regarding the validity of the MRF assumptions for this data set, the locality or "Markov" part virtually always holds if we select a large enough neighborhood size l. We use CV within the supervised learning model fitting procedure to determine how large the neighborhood should be.

The value of *l* that minimizes the CV error corresponds to the neighborhood size that includes all neighboring pixels that serve as useful predictor variables. In this respect, CV identifies the neighborhood size that is required to make the stochastic surface Markov, which follows trivially by definition of the Markov property. In addition, the in-control images in this example passed the stationarity test for textured images of Taylor, Eckley, and Nunes (2014).

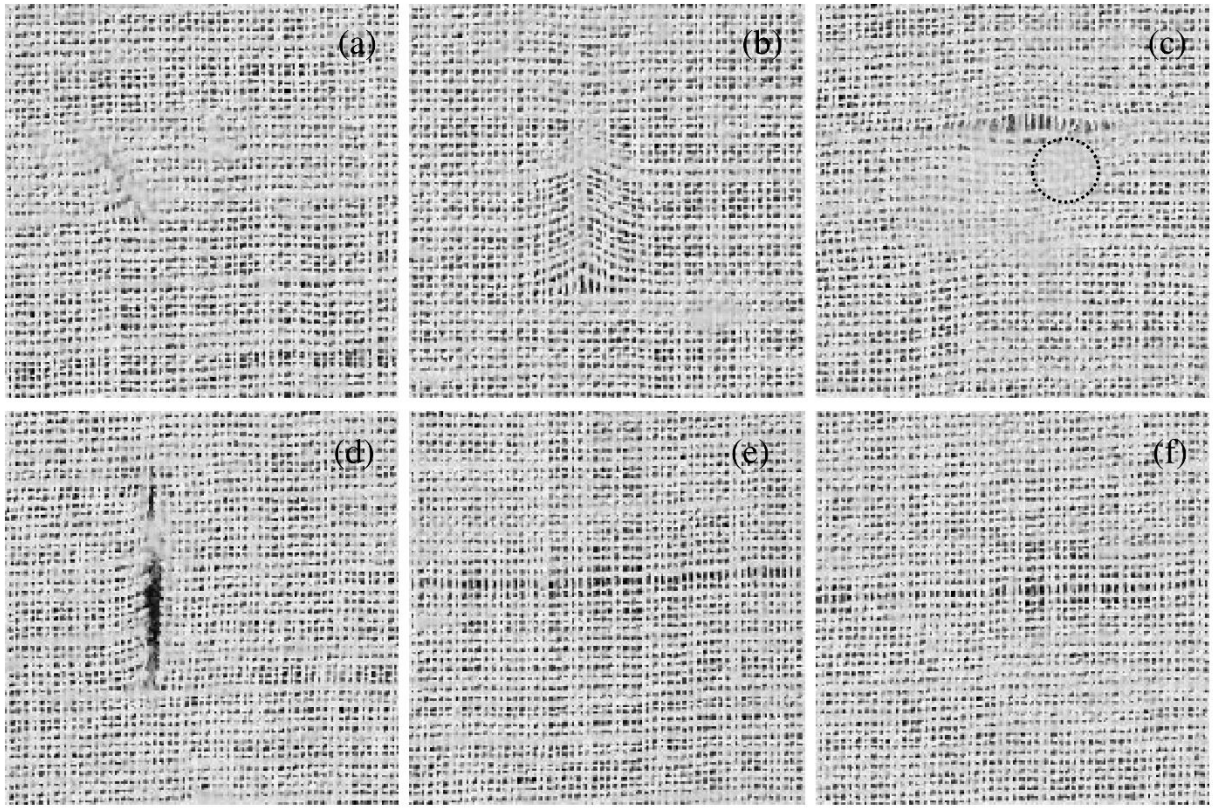


Figure 2.8. Defect-containing textile images corresponding to the in-control image in Figure 2.1(a), but with defects: (a) scratch, (b) fiber direction change, (c) tear, (d) hole, and (e—f) runs. The circled blurry region in panel (c) is an artifact of our imaging system and not a created defect, but it is more severe than the typical imaging blurs.

2.6.1 Monitoring Stage

We used a regression tree as the supervised learning model in this example, and we used the image of size 500×500 pixels shown in Figure 2.1(a) to fit the regression tree. The neighborhood size l of 15 was chosen by CV. From the fitted regression tree, we computed the SMS values and the monitoring statistic S in (2.4) for $N = 94$ Phase I images, for both the A-D-based and B-P-based SMSs, using w = 5, 15, and 25. For the A-D-based statistic, we chose $q_l \approx q_u \approx 1.8 \times 10^{-3}$ (corresponding to 400 observations in each tail), and $p \approx 2.2 \times 10^{-5}$ (corresponding to $r_p = -3.03$ and $r_{1-p} = 3.16$).

For this example, we also compare our algorithm with the EPWMA and EPWMV approaches and the modified version of Lin (2007a) described in Section 2.5. For all methods, we set the control limits based on the empirical distribution of their monitoring statistics, computed from the set of $N = 94$ Phase I images, such that one of the 94 image statistics fell outside the control limits. For the EPWMA method, the control limits (LCL, UCL) were taken to be symmetric about the center line (CL), and likewise for the square root of the EPMVA statistic.

Table 2.3 reports the values of the CLs, LCLs, and UCLs and the monitoring statistics computed for the 6 defect-containing images in Figure 2.8 (which represent Phase II images) for all methods. The statistics that are beyond the control limits are in bold font. As was the case for the simulation example in Table 2.1, the monitoring performance of our algorithm when using the B-P-based statistic is slightly better than that when using the A-D-based statistic in this example. Nevertheless, with either monitoring statistic, the monitoring performance of our algorithm is clearly better than the performance of the other algorithms. As can be seen in Table 2.3, the EPWMA approach does not sound an alarm for any of the six defect-containing images in Figure 2.8. The Lin (2007a) algorithm only sounds an alarm for the most extreme defect in the image in Figure 2.8(d), whereas the EPWMV approach sounds an additional alarm for the image in Figure 2.8(f). In contrast, for most window sizes, our algorithm sounds an alarm for all six defect-

containing images, and the signals often exceed the control limit by a large margin.

Table 2.3. Comparison of monitoring results in the textile example for our approach, and the EPWMA, EPWMV, and Lin (2007a) approaches for $\alpha = 1/94$. The last six columns show the monitoring statistics computed for the six defect-containing images in Figure 2.8. Bold numbers indicate alarmed cases.

Methods		LCL	CL	UCL	(a)	(b)	(c)	(d)	(e)	(f)
A-D	w = 5	–∞	16.0	26.0	**36.7**	**37.1**	**49.1**	**167**	**28.1**	**26.1**
	w = 15	–∞	18.8	29.1	**37.0**	**56.1**	**56.2**	**453**	**34.5**	**34.4**
	w = 25	–∞	23.3	40.3	**45.0**	**103.1**	**70.1**	**376**	36.5	40.2
B-P	w = 5	–∞	9.38	21.2	19.7	**64.0**	**51.9**	**1346**	**27.4**	**29.7**
	w = 15	–∞	1.46	2.7	**4.72**	**33.2**	**12.2**	**1730**	**5.87**	**7.17**
	w = 25	–∞	1.03	1.8	**2.52**	**25.7**	**6.63**	**935**	**3.95**	**4.25**
EPWMA	w = 5	0.29	0.33	0.37	0.33	0.34	0.32	0.31	0.34	0.33
	w = 15	0.06	0.12	0.18	0.13	0.11	0.13	0.10	0.10	0.10
	w = 25	0.04	0.08	0.12	0.09	0.08	0.11	0.07	0.07	0.07
EPWMV	w = 5	2.93	3.52	4.17	3.62	3.51	**4.19**	3.38	3.20	3.77
	w = 15	1.72	2.08	2.48	2.22	2.28	2.27	2.43	2.29	2.40
	w = 25	1.44	1.77	2.13	1.89	1.78	1.92	**2.60**	1.89	**2.15**
Lin (2007a)		–∞	30.9	43.8	28.3	28.2	29.7	**60.7**	32.9	38.3

2.6.2 Diagnostic Stage

In the following, we discuss the diagnostic results of our approach for the example in Section 2.6.1. As in the simulation example in Section 2.5, we used $n_D = 10$ to set the diagnostic threshold for all cases. The diagnostic images for the defect-containing images in Figures 2.7(a—e) are shown in Figure 2.9. The defect and diagnostic results for Figure 2.8(f) are similar to those for Figure 2.8(e), so we omit it here. The first and second rows of Figure 2.9 show the diagnostic images of our algorithm using the A-D-based statistic with w = 5 and 25, respectively. Similarly, the third and fourth rows of Figure 2.9 show the diagnostic images of our algorithm using the B-P-type statistic with w = 5 and 25, respectively. The results with w = 15 for both statistics, which are somewhat in between the results with w = 5 and w = 25, can be found in the online supplement of Bui and Apley (2018a). The diagnostic images for the EPWMV approach with w = 25 (the w value that provided the best monitoring performance for EPWMV) and for the algorithm in Lin

(2007a), constructed in the same manner as ours (using the same diagnostic threshold method described in Section 2.4.2), are also in the online supplement of Bui and Apley (2018a). In each row of Figure 2.9, the first—last columns are diagnostic images for Figures 2.7(a—e), respectively.

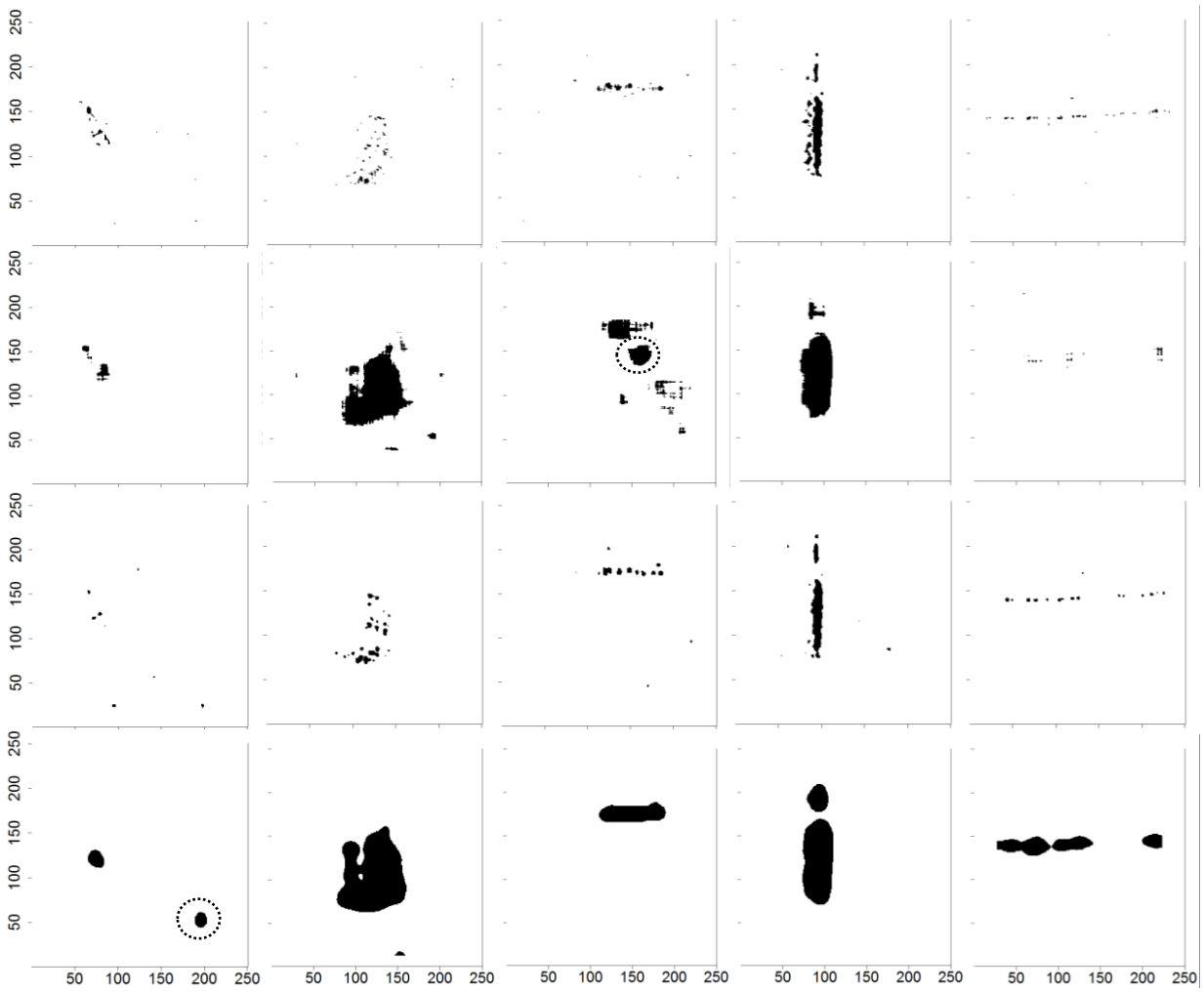


Figure 2.9. Diagnostic images of our algorithm for the defect-containing images in Figure 2.8 using: (1st row) A-D-based statistic with $w = 5$, (2nd row) A-D-based statistic with $w = 25$, (3rd row) B-P-type statistic with $w = 5$, and (4th row) B-P-type statistic with $w = 25$. In each row, the first—last columns correspond to Figures 2.7(a—e), respectively. The circled region is not a defect that we deliberately created, but it may indicate moderately abnormal local behavior in the fabric.

In general, our algorithm has correctly highlighted all the defects that we created, much more

effectively than the EPWMV approach and the Lin (2007a) algorithm have. This is consistent with the comparison of the monitoring performances in Section 2.6.1. For our algorithm, both the A-D-based and B-P-based statistics worked quite well for diagnostic purposes, although B-P-based statistic may provide more pronounced highlights of the defects than the A-D-based statistic does.

There are some highlighted regions in the diagnostic images that do not correspond to the defects that we created, e.g., the circled regions in Figure 2.9. It is important to note that these are not false alarms in the conventional sense, because the purpose of the diagnostic stage is not to sound alarms. None of the highlighted regions that are not associated with the defects we created would have resulted in alarms in the monitoring stage for the control limits that were used for monitoring in Table 2.3, with the exception of the circled regions in Figure 2.9. Although we did not create these as defects, it appears that they are associated with moderately abnormal local behavior of the textile. The region circled in the diagnostic image in the bottom left corner of Figure 2.9 appears to have a relatively loose weave in Figure 2.8(a), and the circled region in the diagnostic image in the middle of the second row of Figure 2.9 is a blurry spot (also circled in Figure 2.8(c)) that is an artifact of our imaging system (which was not an industrial quality system) but that is larger and more pronounced than the typical blurs.

2.7 Conclusions

Stochastic textured surface data have a unique property that precludes the use of the existing SPC methods developed for profile data. Namely, existing profile SPC methods require a gold standard profile or at least a well-defined profile mean with meaningful features. On the other hand, most existing SPC methods applicable to stochastic textured surfaces seek to identify predefined features, which lack generality and are problem-specific by definition, requiring users' knowledge of the defects that are likely to occur. In contrast, we have developed a more general approach that is intended to detect any arbitrary local deviation from the normal in-control statistical behavior of the stochastic textured surfaces.

Our approach uses any appropriate off-the-shelf supervised learning algorithm to characterize the normal in-control statistical behavior of the stochastic textured surfaces. Based on the residuals of the fitted supervised learning model, we proposed two SMSs (A-D-based and B-P-based statistics) to quantify the local behavior of the residuals. We then use the max of the SMSs computed for all pixels in an image as the individual monitoring statistic for that image. We have illustrated the approach with examples of simulated stochastic textured surfaces and real textile fabric images. Both the A-D-based and the B-P-based statistics quite successfully detected and revealed (via the diagnostic images) the existence of defects of various natures. We have observed that the B-P-based statistic provided somewhat better performance than the A-D-based statistic in most of the examples.

There are a number of potential avenues to improve the performance of our approach. First, if the defects occur persistently across multiple images, the monitoring performance could likely be improved by accumulating our individual monitoring statistic S_j using an EWMA or CUSUM type statistic, as mentioned in Section 2.3.1. Combining multiple charts, each with a different value of w, may also be useful to detect a wider range of defect sizes. Furthermore, for the B-P-based statistic, it may be useful to use a large value of w over the interior part of the image (larger w typically improves the monitoring performance of the B-P-based chart) and a smaller value of w around the boundary of the images (because smaller w allows the SMSs to be calculated closer to the boundary). Alternative choices of SMS and monitoring statistic, e. g., treating each neighborhood of residuals as a vector and then using some multivariate monitoring statistic on the residual vector as the SMS, could also potentially improve the performance. Likewise, more complex supervised learning models (e.g., boosted trees, random forests, deep neural networks, etc.) may also improve the performance, albeit at the cost of an increase in computational expense. In fact, we have tried boosted trees and neural networks, in addition to regression trees, but were able to achieve only moderate improvement (in terms of CV error of the supervised learning

model) in our examples. This may be because our computational limitations forced us to work with smaller size images and/or terminate the model fitting optimization algorithm early. Finally, the methods in Qiu and Yandell (1997) and Qiu (1998) might be useful for removing noise-related black pixels in the diagnostic images. We leave these for future studies.

CHAPTER 3

Global Change Monitoring for Stochastic Textured Surfaces

3.1 Introduction

As discussed in the Chapter 2, existing profile monitoring literature is not applicable to stochastic textured surfaces because they do not have a gold standard nor a well-defined, meaningful profile mean function. For illustration, Figures 3.1(a)—(c) show images of a textile material at the scales that show weave patterns. Figures 3.1(d)—(f) plot grayscale image realizations of the 2-D stochastic processes in Section 3.6, which resemble the surface roughness data. As can be seen in Figure 3.1, even if two stochastic textured surface samples have exactly the same nature, their images are completely different on a pixel-by-pixel basis and cannot be matched pixel-wise. Said another way, there are an infinite number of images that are completely different pixel-wise, but that share exactly the same stochastic nature. There is no meaningful gold standard profile for the stochastic textured surfaces in Figure 3.1, nor can one be obtained by alignment, transformation, or warping the image, partly because there are no well-defined features to align. One might naively consider taking the gold standard profile for this type of stochastic textured surface to be the spatial mean function averaged across a collection of images of the same size, but this would be an uninteresting constant function (by the stationarity property, discussed later) and not particularly useful for process control purposes. This distinct property of stochastic textured surfaces requires a different monitoring approach that is not based on the notion of a gold standard profile.

Whereas the method in Chapter 2 only detects *local* defects, many texture-related changes are *global* in the sense that they affect the nature of the entire textured surface, although not as severely as the local defects considered in Chapter 2. Examples of such global changes include changes in the volume fraction or in the nearest neighbor dispersion of particle inclusions in material microstructure images, changes in orientation or thickness (or more generally, in the pattern) of

the textile fibers, etc. To illustrate, Figure 3.1(c) shows an image that was digitally contracted in the horizontal direction by 10%, relative to in-control images of the same textile material as the ones in Figures 3.1(a) and (b). The contracted images constitute a global change in the stochastic nature of the surface, by which the horizontal spacing between the fibers in the weave become smaller on average.

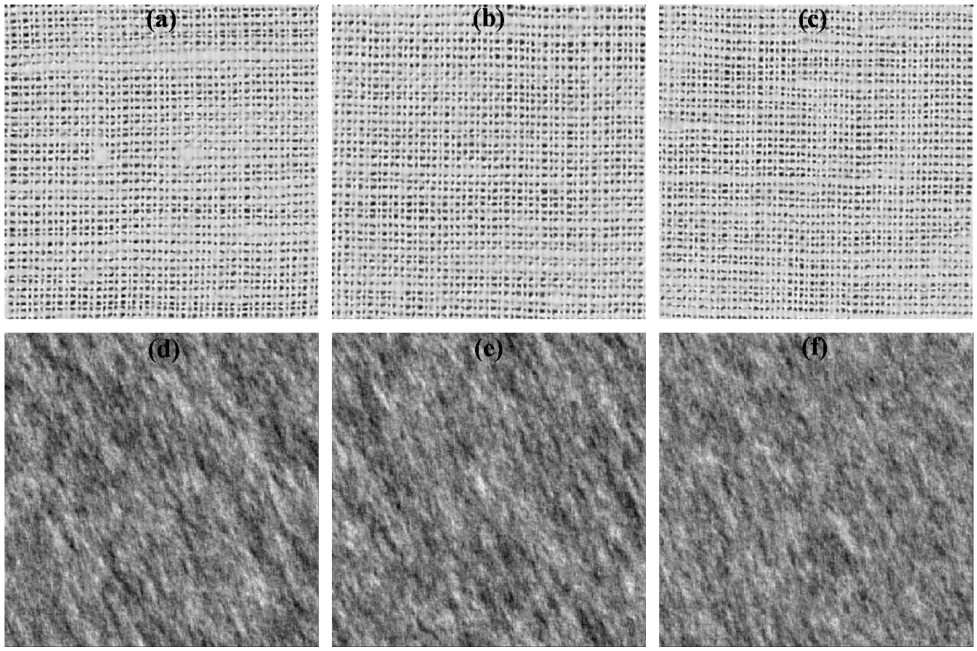


Figure 3.1. Textile and simulated image samples illustrating stochastic textured surfaces and global changes in their nature. Panels (a) and (b) are images of two different locations on a textile swatch under normal and stable manufacturing conditions. Their fibers vary stochastically over space with no distinct geometric features other than the stochastic nature of the fiber patterns. Panel (c) is an image of the same textile material as in panels (a) and (b), but with the weave pattern contracted by 10%, which represents a global change in the stochastic nature of the surface. Similarly, Panels (d) and (e) are images of two realizations of the 2-D stochastic process in Section 3.6 that resemble surface roughness data under normal behavior. Panel (f) is an image realization of a similar 2-D stochastic process but with parameters slightly changed from the normal ones, representing a global change in the surface nature.

Similarly, Figure 3.1(f) shows a simulated realization (as a grayscale image) of a 2-D stochastic process with parameters changed slightly from those of the 2-D stochastic process that was used

to simulate the realizations in Figures 3.1(d) and (e). The change is manifested as a change in the strength and the orientation of the spatial autocorrelation. The details of these processes and their parameters are provided in Section 3.6. Supposing the stochastic process realizations Figures 3.1(d) and (e) represent in-control behavior, the realization in Figure 3.1(f) is the result of a global change in the stochastic nature of the surface.

In this chapter, our goal is to develop a control charting approach that monitors for general global changes in the nature of stochastic textured surface data. We do not assume any prior knowledge of the changes, nor must the changes be persistent across different images (because our approach is akin to a Shewhart individual chart with one charted statistic per image). As in Chapter 2, we use a supervised learning model to implicitly characterize the joint distribution of the pixel values in the image, which provides an implicit characterization of the nature of the stochastic textured surface. Although the underlying supervised learning-based approach for characterizing the stochastic nature of the surface is the same, we focus on detecting a fundamentally different type of change than what Chapter 2 considered. The work in this chapter applies to detecting quite general changes that affect the entire nature of the stochastic textured surfaces but that are very mild in any small local region (e.g., a slight change in the weave pattern of a textile). In contrast, the method in Chapter 2 applies to detecting a very different type of change that is spatially-localized, severe anomalies or defects (e.g., a small hole or tear in the fabric). The distinction between small but severe localized anomalies versus mild global changes in the stochastic nature is not just a minor distinction that requires only minor modifications in the approach. Beyond the common ingredient of representing the surface by a supervised learning model, the approaches for local versus global change detection are completely different. One major difference is that in this chapter we represent each surface (in both Phase I and Phase II) by its own fitted model (to represent the stochastic nature of each surface sample, which could change from sample to sample). Our approach is based on the recognition that a global change in the nature of the surface

corresponds to a change in the joint distribution of the pixels of the image that represents the surface. Because of the high-dimensionality of the problem (images have a large number of pixels), detecting general global changes is a challenging problem. Taking advantage of the compact supervised-learning-based representation of the joint distribution, we develop a monitoring statistic based on generalized likelihood ratio testing (GLRT) principles that measures the extent of the differences in the stochastic nature of the surfaces, based on the differences in their fitted supervised learning models. In contrast, Chapter 2 used a single model to represent the stochastic nature of every surface (since the stochastic nature was assumed the same for each image, aside from possible localized defects), and they monitored the residuals of this single model to detect small local defects. We view the work in this chapter and Chapter 2 as complementary tools that can be used in conjunction with each other, as they are designed to detect completely different types of changes.

The remainder of this chapter is organized as follows. In Section 3.2, we describe how the joint distribution representing the spatial statistical behavior of the stochastic textured surfaces can be approximated via supervised learning, and we give the basic rationale behind our proposed approach. We develop our monitoring statistic and the procedure for constructing the control limit in Section 3.3. Section 3.4 illustrates and evaluates the approach with images of a real textile material. We compare our generic global change detection approach with feature-based approaches for detecting specific changes in Section 3.5. We study the change detection power of our approach with a simulation study in Section 3.6. Section 3.7 concludes the chapter.

3.2 Representing the Joint Distribution and Rationale behind the Approach

As in Section 2.1, suppose we have a sample of N greyscale images, each of which has M pixels. Let $\mathbf{Y}_j = \left[y_{j,1},\ y_{j,2},\ \ldots, y_{j,M}\right]^T$ (j = 1, 2, . . ., N) denote the set of ordered pixel intensities (following a sequence of left-to-right raster scan, moving from the top row to the bottom row of the image $\mathbf{Y}_j$) for the jth image in the sample. Dropping the image index subscript j, let $f(\mathbf{Y})$ denote

the joint distribution of **Y**, which in theory represents the complete statistical behavior of the surface images. Chapter 2 uses the supervised-learning-based technique to characterize the normal, in-control behavior of stochastic textured surface greyscale images for local defect detection in an SPC context. In this chapter, we also use this method to learn the joint distribution $f(\mathbf{Y})$ of the image **Y** via the general regression model in (2.2). In addition, throughout in this chapter, we treat ε_i as if it follows an independent Gaussian distribution, denoted $NID(0, \sigma^2)$, in which case $f\left(y_i|\mathbf{y}^{(i)}\right)$ is the $N\left(g\left(\mathbf{y}^{(i)}\right), \sigma^2\right)$ distribution. From this and (2.1), the joint distribution $f(\mathbf{Y})$ of **Y** is

$$f(\mathbf{Y}) \approx \prod_{i=1}^{M} f\left(y_i|\mathbf{y}^{(i)}\right) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{\left(y_i - g\left(\mathbf{y}^{(i)}\right)\right)^2}{2\sigma^2}\right\}$$

$$= \frac{1}{(2\pi)^{M/2}\sigma^M} \exp\left\{-\frac{\sum_{i=1}^{M}\left(y_i - g\left(\mathbf{y}^{(i)}\right)\right)^2}{2\sigma^2}\right\}. \quad (3.1)$$

From (3.1), we can view the joint distribution of **Y** as characterized by two quantities: (*i*) the model $g(\cdot)$ for predicting a pixel value as a function of its neighborhood pixel values and (*ii*) the prediction error variance σ^2. Both quantities can be estimated by fitting a supervised learning model to the image **Y**. To accomplish this, we first construct a training data set as mentioned briefly above: The *i*th row of this data set consists of the response variable y_i (in the first column) and the predictor variables $\mathbf{y}^{(i)}$ (in the remaining columns), corresponding to a particular response pixel *i* in the image. Similar to Chapter 2, we define the neighborhood by a parameter *l*, which is the number of pixels to the right of, to the left of, and above the response pixel, corresponding to the left-right then up-down raster scan order. As an example, a neighborhood with $l = 2$ is shown in Figure 2.2. The value of *l* should be chosen large enough to include all important predictor variables, i.e., so that the MRF locality assumption holds, which can be accomplished via CV. It should be noted that some pixels near the boundary of the image (e.g., the pixels in the first row of the image in Figure 2.2) will have missing values for $\mathbf{y}^{(i)}$. We exclude any such pixels with

incomplete neighborhoods from the training data. Hereafter let *M* denote the number of training rows, i.e., the number of pixels *i* in the image having complete neighborhoods $\mathbf{y}^{(i)}$.

The rationale behind our approach for detecting global changes in the nature of the stochastic textured surfaces is as follow. Based on the preceding arguments, changes in the nature of the surfaces are equivalent to changes in the joint distribution $f(\mathbf{Y})$, which in turn is equivalent to changes in the pair $\{g(\cdot), \sigma^2\}$ that represents the joint distribution. Consequently, we fit a supervised learning model to an in-control image (or a set of in-control images) to obtain $\{g_0(\cdot), \sigma_0^2\}$, where the subscript 0 indicates that the supervised learning model and error variance characterize the in-control nature. If $\mathbf{Y}_j$ denotes the *j*th newly manufactured stochastic textured surface sample (i.e., the *j*th "on-line" image) to be monitored, we fit a supervised learning model to $\mathbf{Y}_j$ to obtain $\{\hat{g}_j(\cdot), \hat{\sigma}_j^2\}$. Then, in a manner to be developed and explained shortly, we compare the joint distribution implicitly represented by $\{\hat{g}_j(\cdot), \hat{\sigma}_j^2\}$ to the corresponding in-control distribution represented by $\{g_0(\cdot), \sigma_0^2\}$. If the resulting difference between the distributions is large enough, this is an indication that the stochastic nature of image $\mathbf{Y}_j$ has changed from its in-control state. How to compare the implicitly represented joint distributions and quantitatively measure their differences, to serve as a control chart monitoring statistic, is nontrivial. Section 3.3 describes our proposed approach for accomplishing this.

Remark 1: A pixel will generally still be dependent on the pixels after it, even conditioned on the pixels before it. This is analogous to what happens in ARMA time series modeling, in which one predicts an observation y_i given the observations before it. This is done even though y_i is not conditionally independent of the observations after it, given the observations before it. In fact, including the observations after y_i will further improve its prediction (although it would no longer be a future prediction). In the supervised learning model for predicting y_i, we could have included the pixels after y_i as additional predictors, in addition to the pixels before it. We tried this approach but found that the overall monitoring performance of the current approach was better, even though

including predictors after y_i gave smaller prediction errors. We suspect that it may have to do with the residual spatial correlation being smaller when the neighborhood of predictor pixels are the pixels before, but not after, y_i. Going back to the ARMA time series analogy, if we predict y_i given the observations before it (and use the correct ARMA model), then the residuals are temporally uncorrelated. However, if we predict y_i given the observations both before and after it, then the residuals are no longer temporally uncorrelated, even though they have smaller variance than when only using the observations before y_i. In addition, the current approach is more conceptually consistent with the factorization in Section 2.2. There is no analogous factorization when the conditional distributions are conditioned on all other pixels both before and after y_i, although there is a type of Gibbs sampling interpretation.

3.3 Control Chart Construction

In this section, we propose a control chart for monitoring for general global changes in the nature of stochastic textured surfaces. In Section 3.1.1, we develop the monitoring statistic to measure deviations between the implicitly characterized joint distributions for each on-line image $\mathbf{Y}_j$ versus for in-control images, based on GLRT principles. In Section 3.1.2, we discuss the procedure for selecting the control limit, which is based on the monitoring statistics computed for a set of in-control Phase I images.

3.3.1 A GLRT-based Monitoring Statistic

Recall that the underlying joint distribution for normal, in-control stochastic textured surfaces can be represented by $\{g_0(\cdot), \sigma_0^2\}$ based on (3.1). Similarly, the joint distribution for $\mathbf{Y}_j$ can be represented by the analogous quantity $\{g_j(\cdot), \sigma_j^2\}$ for the jth image, an estimate of which is the model $\{\hat{g}_j(\cdot), \hat{\sigma}_j^2\}$ fitted to $\mathbf{Y}_j$. Note that, by definition, a global change causes a deviation of $\{g_j(\cdot), \sigma_j^2\}$ from $\{g_0(\cdot), \sigma_0^2\}$. Thus, we can think of the monitoring goal as testing the following hypotheses for each on-line image j (j = 1, 2, ...):

H_0: $g_j(\cdot) = g_0(\cdot)$ and $\sigma_j = \sigma_0$

H_1: $g_j(\cdot) \neq g_0(\cdot)$ or $\sigma_j \neq \sigma_0$ (3.2)

Throughout, to simplify the development, we treat the in-control distribution (i.e., $\{g_0(\cdot), \sigma_0^2\}$) as known, although it is estimated/fitted to an in-control training image(s). Treating the in-control distribution as known is common in other control charting contexts and may be reasonable if the size of the in-control image sample(s) to which the supervised learning model $g_0(\cdot)$ is fitted is large. Regardless, based on how the control limit is calculated empirically in Section 3.1.2, any uncertainty in $g_0(\cdot)$ is automatically and appropriately reflected in the control limit. By virtue of this, users should make sure that the in-control training image(s) is representative of the surface under normal and stable manufacturing conditions. Note that the training sample can be as large or as small as desired, and can be comprised of pixels from a number of different images.

From (3.1), for a given $\{g(\cdot), \sigma^2\}$, the log-likelihood for $\mathbf{Y}_j$ is

$$l\left(\mathbf{Y}_j; g(\cdot), \sigma^2\right) = -\frac{M_j}{2}\log(2\pi) - \frac{M_j}{2}\log\sigma^2 - \frac{\sum_{i=1}^{M_j}\left(y_{j,i} - g\left(\mathbf{y}_j^{(i)}\right)\right)^2}{2\sigma^2}, \quad (3.3)$$

where M_j is the number of pixels in $\mathbf{Y}_j$ having a full neighborhood. In (3.3) we have included the image index subscript j, so that $y_{j,i}$ and $\mathbf{y}_j^{(i)}$ denote the ith pixel of image $\mathbf{Y}_j$ and the set of neighborhood pixels for $y_{j,i}$, respectively, and $g\left(\mathbf{y}_j^{(i)}\right)$ denotes the predicted value for $y_{j,i}$ using the supervised learning model $g(\cdot)$. Thus, the log-likelihood for $\mathbf{Y}_j$ under H_0 is

$$l\left(\mathbf{Y}_j; g_0(\cdot), \sigma_0^2\right) = -\frac{M_j}{2}\log(2\pi) - \frac{M_j}{2}\log\sigma_0^2 - \frac{\sum_{i=1}^{M_j}\left(y_{j,i} - g_0\left(\mathbf{y}_j^{(i)}\right)\right)^2}{2\sigma_0^2}. \quad (3.4)$$

Assuming the model fitting criterion for the supervised learner is nonlinear least squares (with some complexity or regularization penalties), we can view $\hat{g}_j(\cdot)$ as the value of $g(\cdot)$ that maximizes the log-likelihood (3.3) for $\mathbf{Y}_j$ under H_1. Define the error sums of squares for $g_0(\cdot)$ and for $\hat{g}_j(\cdot)$ applied to $\mathbf{Y}_j$ as $SSE_{0,j} = \sum_{i=1}^{M_j}\left[y_{j,i} - g_0\left(\mathbf{y}_j^{(i)}\right)\right]^2$ and $SSE_{j,j} = \sum_{i=1}^{M_j}\left[y_{j,i} - \hat{g}_j\left(\mathbf{y}_j^{(i)}\right)\right]^2$,

respectively. In (3.3), if we set $g(\cdot) = \hat{g}_j(\cdot)$ and set σ^2 as its MLE

$$\hat{\sigma}_j^2 = \frac{SSE_{j,j}}{M_j} \tag{3.5}$$

under H_1, then the maximized log-likelihood under H_1 is

$$l\left(\mathbf{Y}_j; \hat{g}_j(\cdot), \hat{\sigma}_j^2\right) = -\frac{M_j}{2}\log(2\pi) - \frac{M_j}{2}\log\hat{\sigma}_j^2 - \frac{\sum_{i=1}^{M_j}\left(y_{j,i} - \hat{g}_j\left(\mathbf{y}_j^{(i)}\right)\right)^2}{2\hat{\sigma}_j^2}. \tag{3.6}$$

Using (3.4) and (3.6), our proposed GLRT-based monitoring statistic is

$$\begin{aligned} L_j &= -\frac{2}{M_j}\left(l\left(\mathbf{Y}_j; g_0(\cdot), \sigma_0^2\right) - l\left(\mathbf{Y}_j; \hat{g}_j(\cdot), \hat{\sigma}_j^2\right)\right) \\ &= \log\sigma_0^2 + \frac{SSE_{0,j}}{M_j\sigma_0^2} - \log\hat{\sigma}_j^2 - \frac{SSE_{j,j}}{M_j\hat{\sigma}_j^2} \\ &= \log\frac{\sigma_0^2}{\hat{\sigma}_j^2} + \frac{\hat{\sigma}_{0,j}^2}{\sigma_0^2} - 1, \end{aligned} \tag{3.7}$$

where we have denoted $\hat{\sigma}_{0,j}^2 = M_j^{-1}SSE_{0,j}$ in analogy with (3.5).

We refer to (3.7) as the GLRT monitoring statistic. By the nature of GLRTs, the control chart is one-sided and sounds an alarm that the nature of the surface has changed if L_j exceeds some (upper) control limit.

Remark 2: If the nature of the *j*th image has not changed from the in-control state, then we would expect $\hat{\sigma}_j^2 \approx \hat{\sigma}_{0,j}^2 \approx \sigma_0^2$, so that $L_j \approx 0$. Conversely, if the nature of the *j*th image does change from the in-control state, then we expect a larger L_j. This is easier to see if we take a first-order Taylor approximation of the first term in (3.7) about $\hat{\sigma}_j^2 \approx \sigma_0^2$, which gives

$$L_j = \log\frac{\sigma_0^2}{\hat{\sigma}_j^2} + \frac{\hat{\sigma}_{0,j}^2}{\sigma_0^2} - 1 \approx -\left(\frac{\hat{\sigma}_j^2 - \sigma_0^2}{\sigma_0^2}\right) + \frac{\hat{\sigma}_{0,j}^2}{\sigma_0^2} - 1 = \frac{\hat{\sigma}_{0,j}^2 - \hat{\sigma}_j^2}{\sigma_0^2}. \tag{3.8}$$

If the nature of the image $\mathbf{Y}_j$ changes from the in-control state, then we would expect the new model $\hat{g}_j(\cdot)$ to fit $\mathbf{Y}_j$ better than the in-control model $g_0(\cdot)$. By definition, this means that $\hat{\sigma}_{0,j}^2 > \hat{\sigma}_j^2$, in which case (3.8) implies that L_j will increase.

Remark 3: In deriving the GLRT monitoring statistic (3.7), we have represented the joint distribution by the pair $\{g(\cdot), \sigma^2\}$, and the GLRT is implicitly designed to detect changes in $g(\cdot)$ and/or σ^2. If, instead, we wanted to test only for changes in $g(\cdot)$, assuming that σ^2 does not change, it is straightforward to derive the resulting GLRT statistic as $\sigma_0^{-2}\left(\hat{\sigma}_{0,j}^2 - \hat{\sigma}_j^2\right)$, which is the same as the Taylor approximation (3.8) of L_j. We prefer the L_j GLRT statistic of (3.7) because it is more general in that it also detects changes in σ^2 that are not associated with a change in $g(\cdot)$. This includes both increases and decreases in σ^2. To see this, suppose σ^2 changes but $g(\cdot)$ does not. Because $g(\cdot)$ does not change, we would expect $\hat{\sigma}_j^2 \approx \hat{\sigma}_{0,j}^2 \approx \sigma^2$ for sufficiently large M_j, in which case (3.7) becomes $L_j \approx \log\frac{\sigma_0^2}{\sigma^2} + \frac{\sigma^2}{\sigma_0^2} - 1$. For a fixed σ_0^2, this achieves its minimum when $\sigma^2 = \sigma_0^2$ and monotonically increases as $|\sigma^2 - \sigma_0^2|$ increases.

Remark 4: The assumption of normal iid residuals will often (perhaps usually) be violated, but we view this assumption as only a convenient means to an end. Namely, the GLRT monitoring statistic (3.7) derived under the normal iid assumption has a conceptually appealing interpretation involving a comparison of the prediction error variances for the different models being compared. The conceptual appeal of the statistic holds even if the residuals are not normal or iid. This view is prevalent in the many other contexts across the vast literature in which GLRTs are derived under the normality and/or iid assumptions. Specifically, it often results in a test statistic whose form is appealing even if the normal iid assumption is violated. As will be demonstrated shortly with a real textile example, our monitoring approach based on the normal iid assumption still performed quite well even though this assumption did not strictly hold. Again, we think the reason is partly because the monitoring statistic (3.7) is intuitively appealing even if the residuals are not normal and iid; and partly because the control limits are calculated empirically as described in the next section, as opposed to being derived theoretically based on the normal iid assumption.

3.3.2 Establishing the (Upper) Control Limit for the GLRT

To establish the control limit for our monitoring statistic (3.7), we need its distribution. By

Wilks's theorem, a GLRT statistic under the null hypothesis generally has an asymptotic chi-squared distribution under certain assumptions and as the number of observations goes to infinity. However, our GLRT statistic (3.7) is based on approximations of the joint distributions via some nonparametric supervised learning model, which is also approximately learned from the data, in addition to being based on the normal iid assumption. Attempting to derive some theoretical distribution for the GLRT statistic to set the control limits would be highly problematic, because the control limits could be far off when the assumptions are violated. To illustrate how the assumptions are likely to be violated, Figure 3.2(a) shows a histogram of residuals from the textile example in Section 3.4. The residual distribution is clearly far from the normal distribution indicated by the dashed curve. Likewise, Figure 3.2(b) plots a 2D image of these residuals, which clearly shows spatial correlation between the residuals. Because of issues like these and the complexity underlying our GLRT statistic, it is doubtful that any useful analytical approximations to its distribution could be derived.

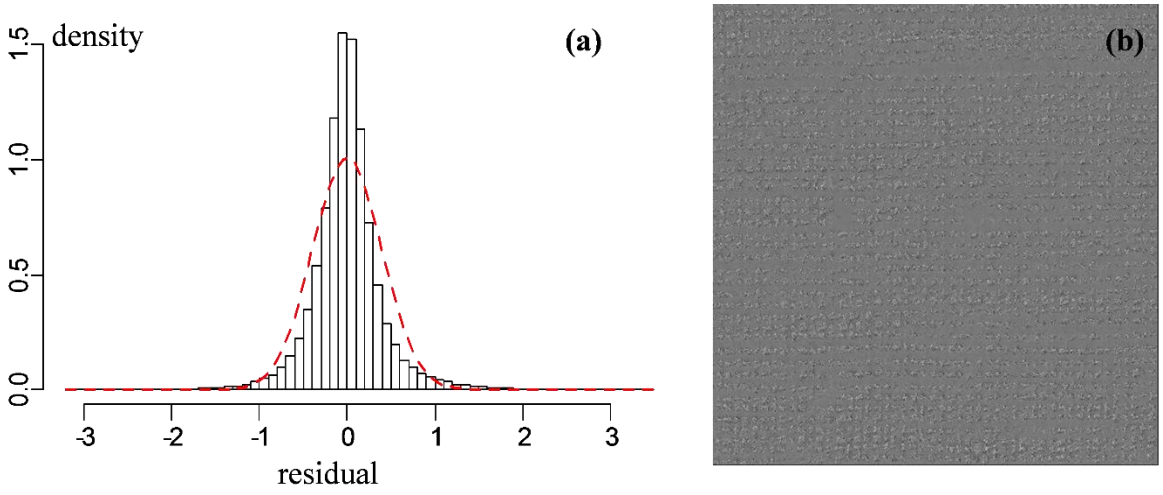


Figure 3.2. (a) Histogram of and (b) image of the residuals for a sample in the textile example of Section 3.4. The dashed curve in Panel (a) is the normal probability density function with mean and standard deviation the same as for the residuals.

In light of this, we recommend establishing the control limits for L_j based on its empirical

distribution over a set of Phase I images. One would first fit the supervised learning model $\{g_0(\cdot), \sigma_0^2\}$ to a sufficiently large training image or a set of training images that are known to be in-control. Then, for a set of additional Phase I images $\{\mathbf{Y}_j : j = 1, 2, \ldots, N\}$ that are believed to be in control, the statistics $\{L_j : j = 1, 2, \ldots, N\}$ are calculated via (3.7), which requires fitting the supervised learning model $\{\hat{g}_j(\cdot), \hat{\sigma}_j^2\}$ for each of the *N* images.

Let α denote the desired false alarm rate of the chart. If the Phase I sample size *N* is large enough, then one can simply use the upper α quantile of the empirical distribution of $\{L_j : j = 1, 2, \ldots, N\}$ as the control limit. However, if the Phase I *N* is too small (e.g., $N < \alpha^{-1}$), then we recommend the following parametric approximation to the empirical distribution of $\{L_j : j = 1, 2, \ldots, N\}$. The approximation assumes that $L = a + b\chi_k^2$, where χ_k^2 denotes a chi-square random variable with *k* d.f., and *a* and *b* are constants. The constants *a* and b and the d.f. *k* are estimated by equating the first three sample moments of $\{L_j : j = 1, 2, \ldots, N\}$ with the theoretical moments of $a + b\chi_k^2$.

Let $\kappa_1 = a + bk$, $\kappa_2 = 2b^2k$, and $\kappa_3 = 8b^3k$ denote the first three central moments of $a + b\chi_k^2$, and let $\hat{\kappa}_1 = N^{-1}\sum_{j=1}^{N} L_j$, $\hat{\kappa}_2 = N^{-1}\sum_{j=1}^{N}(L_j - \hat{\kappa}_1)^2$, and $\hat{\kappa}_3 = N^{-1}\sum_{j=1}^{N}(L_j - \hat{\kappa}_1)^3$ denote the corresponding empirical moments of $\{L_j : j = 1, 2, \ldots, N\}$. Equating the moments gives

$$\begin{cases} a = \hat{\kappa}_1 - {2\hat{\kappa}_2^2}/{\hat{\kappa}_3} \\ b = {\hat{\kappa}_3}/{4\hat{\kappa}_2} \\ k = {8\hat{\kappa}_2^3}/{\hat{\kappa}_3^2} \end{cases} \tag{3.9}$$

If this approximation is used, the control limit can be set at $a + b\chi_{k,\alpha}^2$, where $\chi_{k,\alpha}^2$ denotes the upper α quantile of the chi-squared distribution with k d.f.

3.4 Examples in a Textile Application

3.4.1 Data description and fitting the in-control model

Our Phase I data in this example is a set of 350 real textile fabric images, each of which is

500×500 pixels, taken from non-overlapping areas on the same fabric roll to reduce the possibility of unknown global change being present in the sample. These images passed the test of stationarity for textured images provided in the LS2Wstat R package (Taylor and Nunes 2014). For details of this test, see Taylor et al. (2014). We have released this data set in the R data package **textile2** (Bui and Apley 2017b). The texture of this fabric is quite complex with random thicknesses of and distances between fiber strands, as shown in the images in Figures 3.1(a) and (b). Because the lighting conditions in our laboratory setup may have varied slightly from image to image, to diminish these effects we standardize all images by subtracting from each pixel the average greyscale value for all pixels in that image, and then dividing by the greyscale standard deviation for all pixels in the image.

Note that the textile images in this example are not isotropic, because the stochastic behaviors along the horizontal, vertical, and diagonal directions are all different. Hence, the images should be aligned/registered as best as they can by a rotation as a first step. Otherwise, the stochastic nature of the two misaligned samples will be different even if properly aligned versions share the same nature. Failing to align the images via a rotation would result in either an overly wide control limit (if Phase I images differ from sample-to-sample because of alignment variation) or excessive nuisance alarms (if only Phase II images are misaligned). Alignment for the textile samples can be accomplished approximately, because the threads are approximately horizontally and vertically oriented, and this was done as a pre-processing step. It should be noted that, because of their stochastic nature with no clear and distinct features, the image samples cannot be precisely aligned and can only be aligned in an approximate sense.

We used one of the 350 images, the one shown Figure 3.1(a), as the training image to which we fit $g_0(\cdot)$. The error variance σ_0^2 was taken to be its MLE. We use regression trees as the supervised learning model in this chapter because it is relatively fast to fit (note that we need to fit a new $\hat{g}_j(\cdot)$ for each image *j*). The neighborhood size $l = 15$ was selected via CV. The best values

for any complexity parameters of g_0 (which for trees influences the number of nodes in the fitted trees) were also chosen by CV. In the following, Section 3.4.2 describes the Phase I analysis to find the control limit, and Section 3.4.3 implements the GLRT control charts using this control limit on image samples containing three types of global changes.

3.4.2 Phase I Control Limit Estimation

We performed a Phase I analysis to estimate the control limit using the remaining N = 349 images. For every image, we fit a regression tree $\hat{g}_j(\cdot)$ to the data set constructed from the image with the same neighborhood size l = 15 that was chosen by CV when fitting $\hat{g}_0(\cdot)$ in Section 3.4.1. To avoid performing CV for every image, which is computational expensive we fixed all complexity parameters for each $\hat{g}_j(\cdot)$ to be the same as for $\hat{g}_0(\cdot)$. In other words, we only grew the regression trees $\hat{g}_j(\cdot)$ until they reached the complexity level of $\hat{g}_0(\cdot)$. After fitting $\hat{g}_j(\cdot)$, the GLRT statistic for each image was computed via (3.7). After obtaining the GLRT statistics for all 349 Phase I images, for demonstration, we used the transformed chi-square approximation described in Section 3.1.2 to approximate the sampling distribution of $\{L_j: j = 1, 2, \ldots, 349\}$. The parameters via (3.9) were $a = 0.0529$, $b = 0.0035$, and $k = 15.58$ (non-integer d.f. are allowed in our software) which resulted in a trial control limit of 0.1762 using a false alarm rate α = 0.0027. Figure 3.3 shows the Phase I control chart for the 349 L-statistics using this trial control limit.

The trial control chart signals two observations (circled in Figure 3.3). To investigate these alarms, Figure 3.4 plots the images corresponding to these two signals (Figures 3.3(a) and (b)) together with four other representative images that did not signal (Figures 3.3(c)—(f)). The images in Figures 3.3(a) and (b) correspond to observation #268, which has the largest GLRT statistic, and observation #44, which has the second largest statistic, respectively. The fiber strands in Figure 3.4(a) appear to be much curvier than those of the images that did not signal in Figures 3.3(c)—(f). The same is true in Figure 3.4(b) but to a lesser extent. In fact, these two images were physically adjacent on the same fabric swatch, and this part of the swatch was distorted during the course of

our image acquisition. Hence, we removed these images from the set of Phase I images when recalculating the control limit.

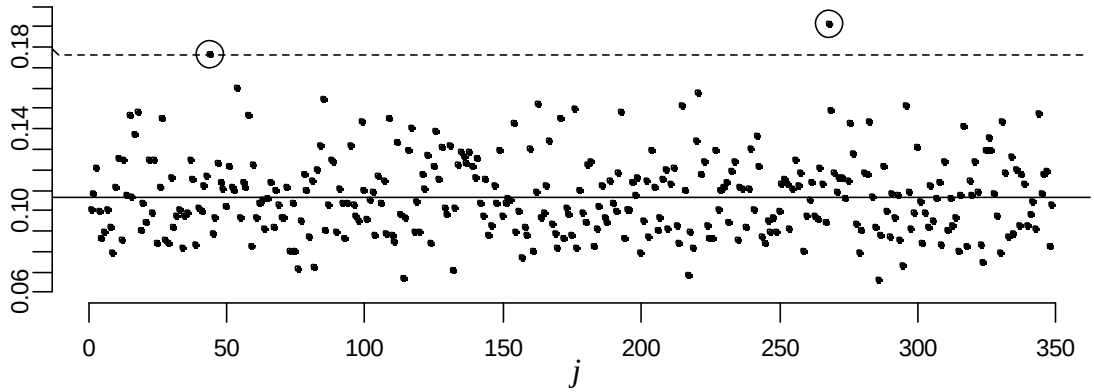


Figure 3.3. Phase I analysis for the textile example using the GLRT monitoring statistic. The vertical axis is the GLRT statistic (3.7) computed for the 349 Phase I images, and the horizontal axis is the image index *j*. The trial control limit (dashed line) was constructed using the transformed chi-square approximation with $\alpha = 0.0027$. Observations #44 and #268 (circled) fell above the trial control limit.

After removing the two images that signaled in Figure 3.3, the remaining 347 Phase I images were used to find the final control limit. For this, we used the empirical distribution $\{L_j : j = 1, 2, \ldots, 347\}$ without the chi-square approximation. The reason we did not use the chi-square approximation is to allow a more common basis for the comparisons in Section 3.5, by using the nonparametric empirical distributions to select the control limits for both algorithms. The final GLRT control limit that we will use in Phase II monitoring in the next section is 0.1573.

3.4.3 Phase II monitoring

In this section, we monitor three sets of Phase II images containing three different types of global changes. We created the changes by digitally modifying a number of images that were originally in-control, coming from the same textile material that was used in Phase I. In the first and second sets, the changes in the Phase II images were created by digitally contracting in-control images in the vertical and horizontal directions, respectively. For each direction, we used four levels of contraction (5%, 10%, 15%, and 20%) and created 25 out-of-control images for each

level. Figure 3.1(c) shows an image with a horizontal contraction level of 10%, which is quite difficult to visually distinguish from the in-control images in Figures 3.1(a) and (b).

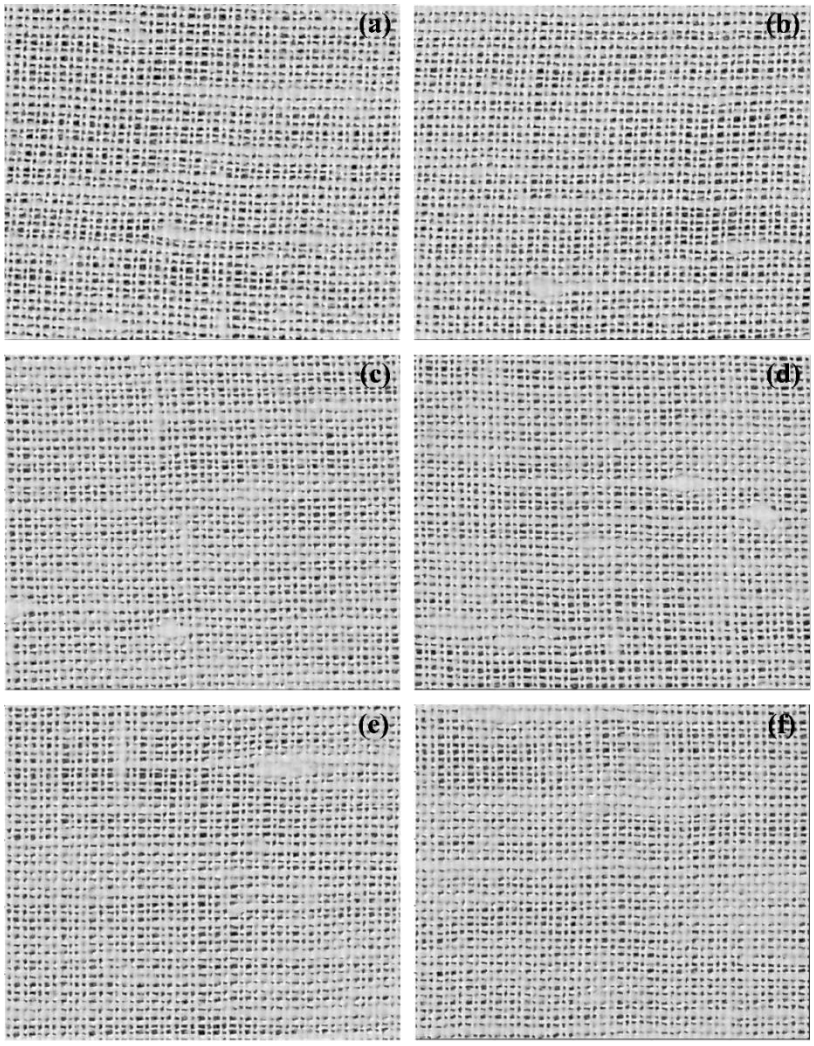


Figure 3.4. Investigating the two signals in the trial control chart of Figure 3.3: (a) image #268, which has the largest GLRT statistic, (b) image #44, which has the second largest statistic, and (c—f) four representative images whose statistics fell below the control limit. The fiber strands in panel (a), and to a lesser extent in panel (b) are curvier than in panels (c—f), which was the cause of the signals.

We refer to the global change that we created in the third set as "matchbox" change, for which Figure 3.5 provides a stylized illustration. If the original image is depicted in Figure 3.5(a), Figure 3.5(b) depicts (an exaggerated version of) the image after the matchbox transformation. After

matchboxing, the vertical lines remain vertical, whereas the horizontal lines are rotated (as how an empty matchbox collapses). We digitally created 143 out-of-control Phase II images having such a matchbox change, and all 143 images were given the same level of matchboxing. Figures 3.5(a) and (b) show an in-control image and its matchboxed version, respectively. Only a small level of matchboxing was introduced, so that the change is virtually indistinguishable by the naked eye (see Figure 3.6).

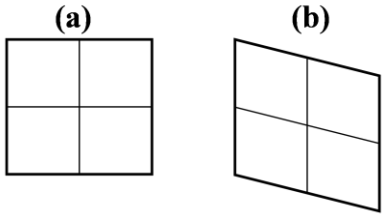


Figure 3.5. Stylized illustration of the matchbox change created for the third set of out-of-control images (the four cells represent four pixels): (a) original in-control image, and (b) image after the matchbox change.

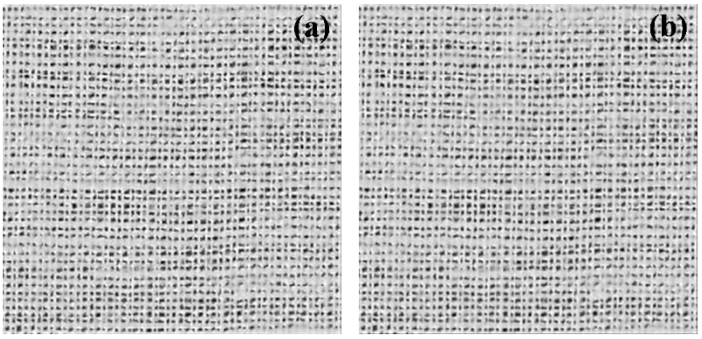


Figure 3.6. Illustration of the matchbox change in Figure 3.5 applied to an image, which is small enough to be virtually visually unrecognizable: (a) original in-control image, and (b) the same image in panel (a) but after adding the matchbox change.

The top left and middle left panels of Figure 3.7 show the monitoring results of our algorithm for the 100 Phase II images containing the vertical and horizontal contraction changes, respectively. For reference, they also show the GLRT statistics for some of the Phase I images, which are the points to the left of the vertical line at index $j = 1$. For $j \geq 1$, the Phase II images are ordered in terms of decreasing level of contraction, and the amount of contraction (20%, 15%,

10%, and 5%) is indicated on the figure. For both the horizontal and vertical contractions, our algorithm consistently signaled individual observations with contraction levels of about 15% or greater. For lower contraction levels, even though the GLRT statistics often fall below the control limit, they mostly lie above the center line of the control chart (indicated by the horizontal solid line). Hence, if such changes were persistent across fabric samples (e.g., caused by a persistent production fault), an EWMA- or CUSUM-type chart on the GLRT statistic could likely accumulate the information across the image samples and detect the changes. The bottom left panel of Figure 3.7 shows the monitoring results of our algorithm for the matchbox change example, which is quite remarkable. Almost every Phase II image containing the matchbox changes caused an alarm. Overall, our algorithm performance is quite impressive considering that it does not incorporate any prior knowledge of the nature of the changes.

3.5 Generic versus Pattern-Specific Monitoring

As previously discussed and demonstrated in the examples in Section 3.4, our approach is completely generic in that it can detect general changes in the nature of the stochastic textured surfaces, and does not require any prior knowledge of the nature of the changes in order to detect them. To illustrate the benefits of being able to detect general changes without advance knowledge of what the changes are, this section compares our approach with a feature-based monitoring algorithm that is designed to detect a specific pattern of change but that fails to detect changes that are not what the algorithm was designed to detect. We use the same textile example as in Section 3.4.

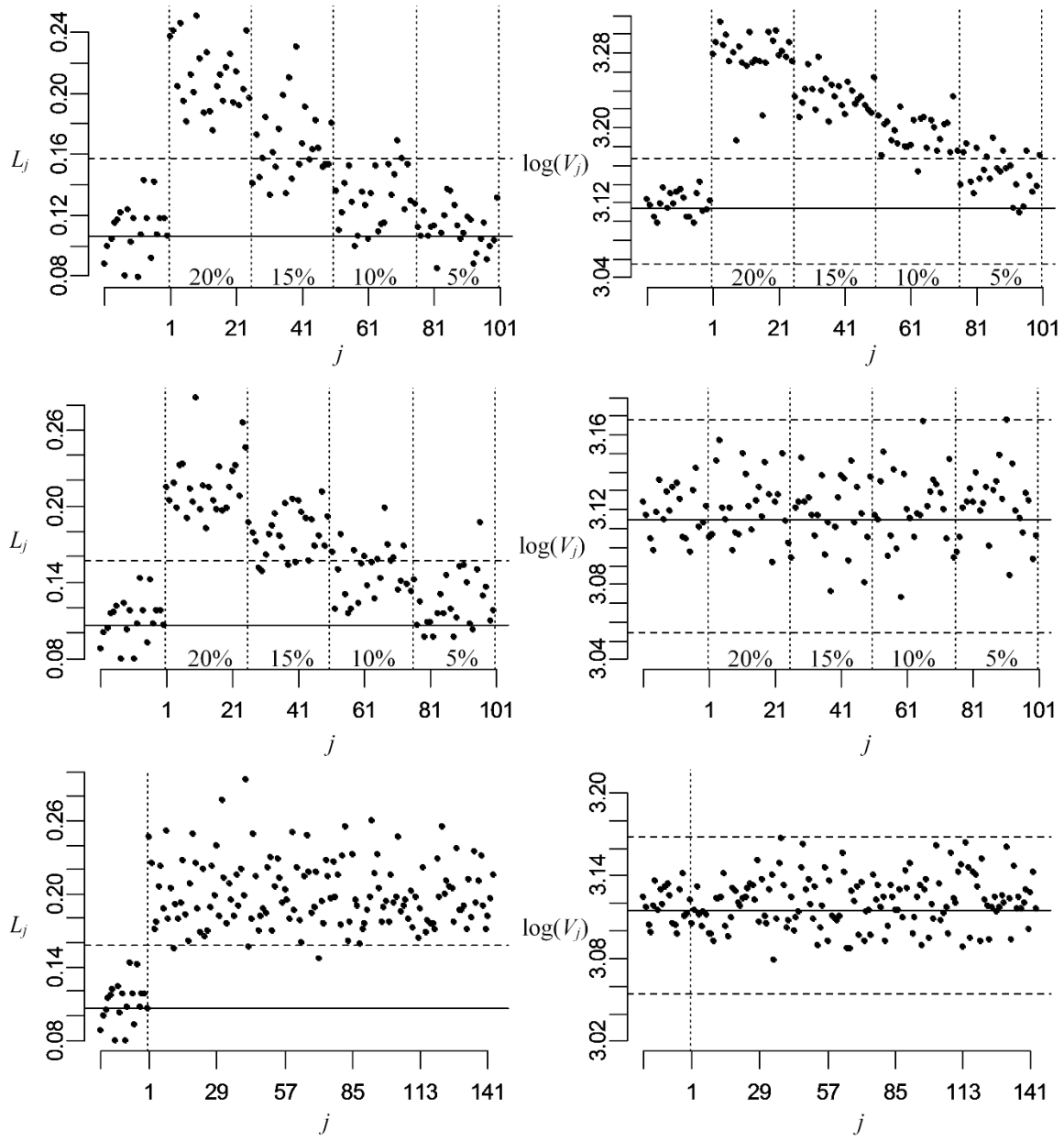


Figure 3.7. Monitoring results for our algorithm (left panels) and for comparison the FFT feature-based algorithm (right panels) for textile examples containing: (top panels) vertical contraction changes, (middle panels) horizontal contraction changes, and (bottom panels) matchbox changes. The vertical line in each panel at index $j = 1$ separates the Phase II images ($j \geq 1$) from some Phase I images $j < 1$ for comparison. The other vertical lines (at indices $j = 25$, 50, 75, and 100) in the top and middle panels group the Phase II images having the same level of contraction, which is indicated by the percentage number in each group. In all panels, the control limit(s) are the horizontal dashed lines, and the center line is the horizontal solid line.

Suppose that we knew in advance that the vertical contraction changes might occur and wanted to design a feature-based algorithm to detect this specific change. For this, we considered the following monitoring algorithm based on a 1-D fast Fourier transforms (FFT) algorithm, which we subsequently refer to as the FFT algorithm. We first perform a 1-D FFT for each column of an image, resulting in an FFT matrix. In the FFT matrix, each row corresponds to a frequency, and the values in that row represent the magnitudes of the corresponding frequency component for each column. We then take the row averages of the FFT matrix, which represent the average magnitudes of the different frequency components in the vertical direction. Finally, as our feature-based FFT monitoring statistic, we use the frequency corresponding to the first peak. We denote this statistic for image j by V_j. Figure 3.8 plots V_j for an in-control image of size 500x500 in the example in Section 3.4. Because of the semi-periodic nature of the fibers, the frequency location for the first peak is closely related to the fiber spacing in the vertical direction, and the subsequent peaks are the higher harmonics. Consequently, monitoring V_j should allow the vertical contractions of the images to be detected.

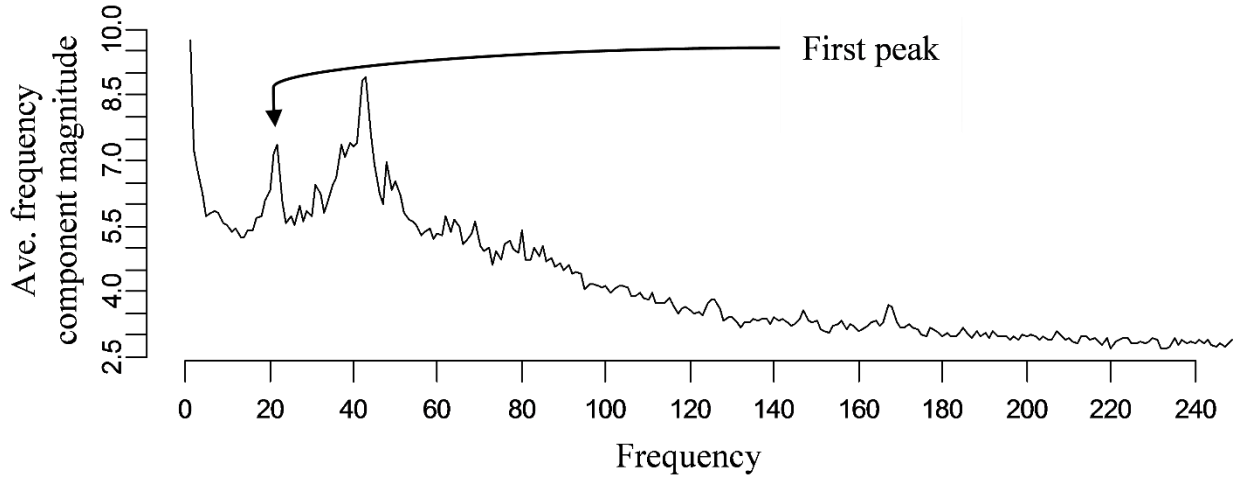


Figure 3.8. Vertical frequency spectrum for an in-control image of size 500x500 illustrating the rationale behind the feature-based FFT algorithm. The vertical axis is the average magnitude (averaged horizontally across the image) of the corresponding frequency component in the vertical direction. The feature-based monitoring statistic is the frequency corresponding to the first peak with positive frequency.

We computed the feature-based FFT monitoring statistic V_j for the 349 Phase I images in Section 3.4.1. The values of V_j for the two Phase I images that were out-of-control for the GLRT statistic in Section 3.4.2 were quite close to the center line for the V_j chart and did not signal. Thus, the feature-based V_j chart failed to detect the abnormal behavior of these two images in the Phase I sample, as this particular abnormal behavior (see Figure 3.4 and the surrounding discussion) differed from the specific vertical contraction changes that the feature-based method was designed to detect.

The results are similar for Phase II monitoring. Based on the empirical distributions of the V_j statistic from the Phase I images, we set the control limits for the V_j chart at the upper and lower $\alpha/2$ quantiles of its empirical distributions with the same α that we used for the GLRT chart in the examples in Section 3.4. The right panels of Figure 3.7 show the monitoring results of the V_j chart for the same three sets of Phase II images that we considered in Section 3.4.3. In the top panel, for which the image contraction changes are in the vertical direction, the V_j chart detects the changes very well and somewhat better than the GLRT chart, although the GLRT chart still detected the vertical contractions quite effectively. This is understandable, because the feature-based V_j chart was specifically designed to detect vertical contractions, whereas the GLRT chart was designed to detect very general changes.

However, the results are completely different in the middle and bottom panels of Figure 3.7, for which the image changes were not the specific changes that the V_j chart was designed to detect. The V_j chart completely fails to detect these changes, for both the horizontal contractions (middle panel) and the matchbox changes (bottom panel). These examples demonstrate the drawback of using a feature-based monitoring algorithm designed to detect only a specific type of change. In contrast, our GLRT algorithm can detect very general changes in the stochastic nature of the textured surfaces without requiring any prior knowledge of the changes.

3.6 Simulation Study

This section studies the monitoring power of our approach via a Monte Carlo simulation example. Let $y(i, k)$ denote the image pixel intensity at pixel location (i, k) where i and k the row and column indices, respectively. Each pixel in the simulated images was generated via the spatial autoregressive model

$$y(i,k) = \phi_1 y(i-1,k) + \phi_2 y(i,k-1) + \varepsilon(i,k) \tag{3.10}$$

where ε is a zero-mean Gaussian white noise. Note that the values of ϕ_1 and ϕ_2 determine the nature of the simulated images. For plotting purposes, all generated processes were translated/rescaled to the interval [0, 255] to acquire the respective greyscale images. Although not required, all simulated images have the same size of 250×250 pixels. We also subsequently standardized all the images as explained in Section 3.4 before applying our algorithm.

For each Monte Carlo replicate, we conducted the following tasks. First, we generated an in-control training image for fitting $\hat{g}_0(\cdot)$ and another N = 1000 in-control images to establish the control limit. We used $\phi_1 = 0.6$ and $\phi_2 = 0.35$ for the in-control process. For example, Figures 3.1(d) and (e) show two image realizations of this in-control process. Out-of-control processes were represented by using parameter values other than these. To study the monitoring power of our approach, we used two mildly out-of-control processes for generating Phase II images. For the first out-of-control process, ϕ_1 and ϕ_2 were reduced by 2% from the in-control values, i.e., $\phi_1 = 0.6 \times 0.98 = 0.588$ and $\phi_2 = 0.35 \times 0.98 = 0.343$. Figure 3.1(f) shows an image realization of this out-of-control process, which looks quite similar to the in-control images in Figures 3.1(d) and (e). Similarly, for the second out-of-control process, ϕ_1 and ϕ_2 were reduced by 5% from the in-control values. We generated 100 Phase II image samples for each out-of-control process. Similar to Section 3.4, we used a regression tree as the supervised learning model for faster model fitting. It is interesting to note that a neighborhood size of l = 1 was obtained via CV, and this value corresponds to the correct value according to Eq. (3.10).

Table 3.1 reports the average power across 10 Monte Carlo replicates for our approach. We used a small number of replicates because each replicate was time-consuming and because the detection power was large enough that 10 replicates were sufficient to draw the below conclusions, considering that there were 100 Phase II images in each replicate. Here we used the empirical distribution of the GLRT statistics computed for the 1000 Phase I images to establish the control limit corresponding to a false alarm rate of 0.003. That is, we set the control limit so that three out of 1000 Phase I images had a GLRT statistic beyond it. Remarkably, Table 3.1 shows that our algorithm correctly signaled almost all out-of-control Phase II images, even the ones with the very mild 2% change in the parameters.

Table 3.1. Average detection power for the simulation example using $\alpha = 0.003$

ϕ_1 and ϕ_2 being reduced by	Power
2%	0.997
5%	1.000

To give an alternative picture of the detection power, Figure 3.9 shows boxplots of $\{L_{r,j}/UCL_r : r = 1, 2, \ldots, 10; j = 1, 2, \ldots, 100\}$, where $L_{r,j}$ is the GLRT statistic of the *j*th Phase II image in the *r*th replicate, and UCL_r is the upper control limit in the *r*th replicate. Note that $L_{r,j}/UCL_r = 2.0$ (for example) means that the GLRT statistic of the *j*th Phase II image in the *r*th replicate exceeded the upper limit in that replicate by a factor of 2.0. Table 3.1 and Figure 3.9 demonstrate that our algorithm can detect the relatively small changes in this example with high probability.

3.7 Conclusions

Stochastic textured surface data do not have gold standards, which renders existing profile SPC algorithms largely inapplicable. The existing image monitoring techniques that could be applied to textured surface monitoring are highly feature-based and problem-specific. Chapter 2 developed a more general stochastic textured surface monitoring approach that does not require prior

knowledge of the types of defects to be detected; however, this method is specifically designed for monitoring local defects that constitute changes in some spatially localized area of the stochastic textured surfaces. In this chapter, we develop a very generic monitoring approach for detecting general global changes in the stochastic nature of the textured surfaces. We used supervised learning to implicitly estimate and characterize the joint distributions that represent the statistical behavior of the stochastic textured surfaces. Based on this characterization of the joint distribution, we developed a monitoring statistic based on GLRT principles to detect general changes in the joint distribution, which is equivalent to detecting general changes in the nature of the stochastic textured surfaces.

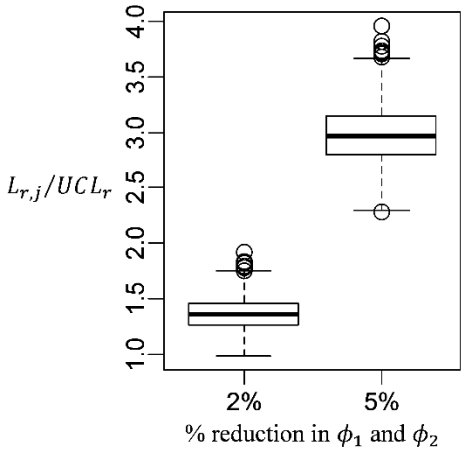


Figure 3.9. Boxplots of $\{L_{r,j}/UCL_r : r = 1, 2, \ldots, 10; j = 1, 2, \ldots, 100\}$, where $L_{r,j}$ is the GLRT statistic of the *j*th Phase II image in the *r*th replicate, and UCL_r is the upper control limit in the *r*th replicate.

We demonstrate the effectiveness of the method using a set of real textile fabric images and a set of simulated images, for which our algorithm successfully detected the global changes without any advance knowledge of the nature of the changes. We also compared our approach to a feature-based algorithm designed to detect specific changes (vertical contractions) in the images for the real textile example, and we showed that our general approach is far superior at detecting changes that differ from the specific ones that the feature-based method was designed to detect. Moreover,

for the specific changes for which the feature-based algorithm was designed, the detection performance of our algorithm was not too far below that of the feature-based algorithm, which we view as remarkable considering that our algorithm can detect general global changes and was not designed for any specific changes. Finally, our approach does not require the global changes to be persistent, because our monitoring statistic L_j is associated with each individual image j, in the spirit of Shewhart individual control charts. However, for situations in which the changes are persistent, we expect that the monitoring performance could be improved by accumulating information across multiple images using EWMA- or CUSUM-type concepts.

We have focused on sounding an alarm (change detection), as opposed to diagnosing the nature of the change. Diagnosing which image(s) experienced a change is quite straightforward: Since our algorithm has one statistic for each image, we can simply look at which images had statistics that exceeded or were close to the control limit. For example, the left panel of Figure 3.7 shows that the contraction change occurred abruptly and then gradually decreased in magnitude, and the matchbox change occurred abruptly and then maintained its magnitude across the set of Phase II samples. Diagnosing the nature of the global change in the stochastic behavior (e.g., that the textile fiber spacing is larger than normal in the horizontal direction) is more challenging than diagnosing the temporal evolution of the change. This would be very useful, however, as understanding the nature of the change would help in identifying the root cause of the change. As future work, we are currently investigating methods to facilitate visualizing the nature of the change. One simple, immediate solution is to plot side-by-side some images before and after the change to allow users to visually discern its nature.

The computational expense of our approach may be prohibitive in some applications. Using a regression tree without CV as the supervised learner (which is not required for the on-line images, because the complexity parameters for each $\hat{g}_j(\cdot)$ can be chosen the same as for $\hat{g}_0(\cdot)$), the current algorithm is relatively fast. For example, it took 0.36 seconds on average to compute the

monitoring statistic for an image in the simulation example in Section 3.6, using an Intel Core i7-2600 CPU with 3.40-GHz processing speed on Windows 7 Professional. However, if using more computationally expensive supervised learning models such as neural networks, boosted trees, random forests, etc., the computational expense may be prohibitively expensive, depending on many factors, including whether the model fitting procedure is stopped early.

CHAPTER 4

Understanding Variation in Stochastic Textured Surfaces

4.1 Introduction

In this chapter, we again focus on stochastic textured surface data that have a stochastic nature. Unlike typical profile or multivariate data whose discretized points or elements correspond one-to-one across observations, image samples of this nature do not match pixel-by-pixel even when the process is operating normally under ideal conditions (i.e., in control conditions using control charting terminology). Hence, we cannot simply subtract one stochastic textured surface image sample from another (or from a transformed or warped version, to match features) to meaningfully represent their difference, and it is not straightforward how to quantify the difference between the pair of surface samples. Hereafter, the term "surface" will refer to a stochastic textured surface, and "image sample" will refer to a sample of such a surface.

The stochastic nature across a set of image samples can vary as a result of manufacturing or other condition changes. To illustrate, Figure 4.1 shows various image samples for the simulated 2-D stochastic process considered in Section 4.4, in which the surface nature is different for each sample. Specifically, each sample was generated from a stochastic process model but with different values of two underlying model parameters (as will be described in more detail later, the two parameters control the rate of decay and the orientation angle of the spatial autocorrelation). The result is that the 16 image samples in Figure 4.1 are not 16 statistically equivalent realizations of the same stochastic process; rather, they are realizations of different stochastic processes and hence have stochastic nature that varies across the set of image samples. The differences across the set of image samples could represent surface roughness behavior that varies across a set of manufactured metal parts or material microstructure characteristics that vary across a set of microstructure samples, due to unstable processing conditions that vary over time. Throughout, we use the term "variation" to refer to such systematic differences in the stochastic nature of the

surfaces across a set of image samples.

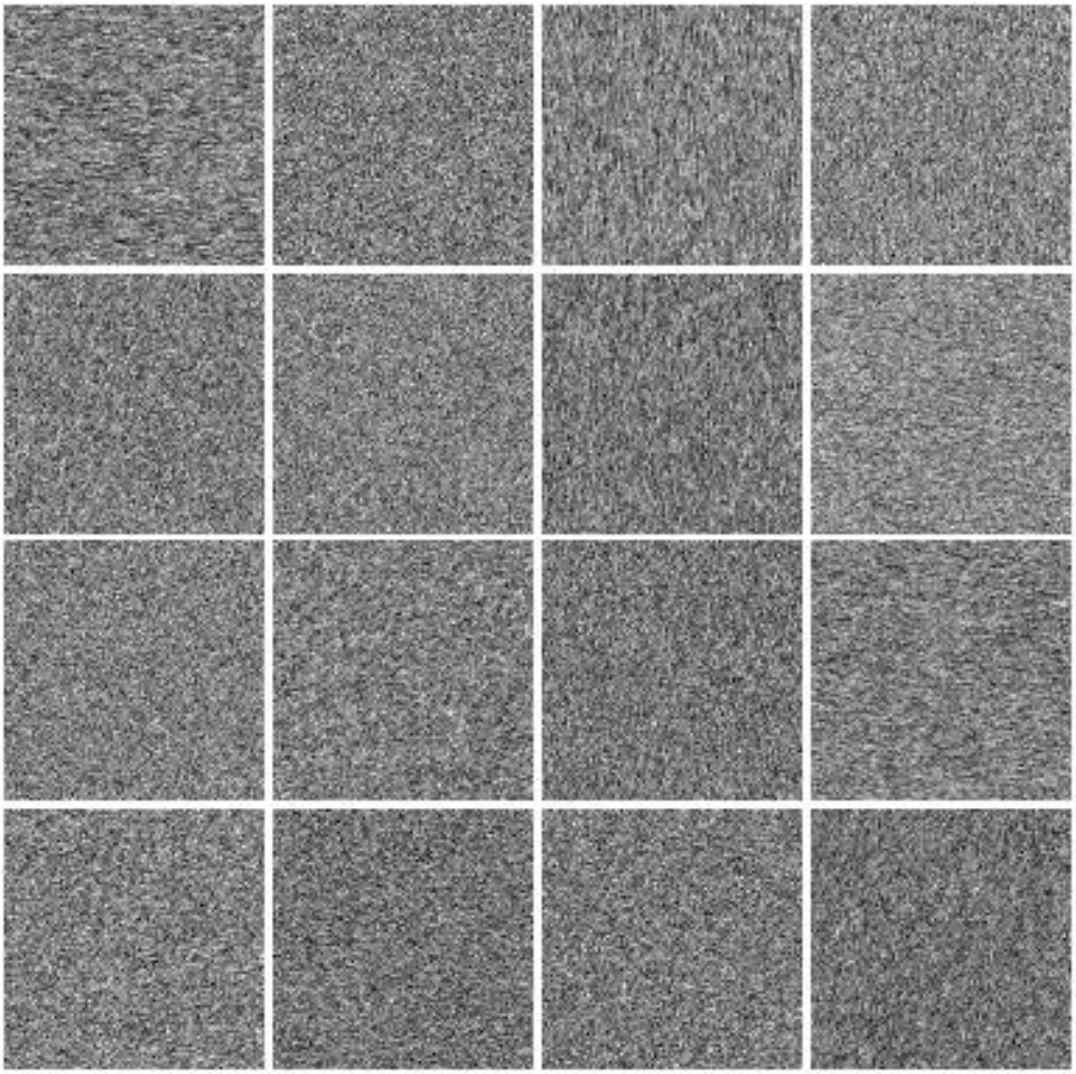


Figure 4.1. A set of 16 image samples, each simulated from the 2-D stochastic process considered in Section 4.4, but with process parameters varying across the set. The varying parameters constitute two systematic variation patterns that represent a changing spatial decay level and a changing orientation angle for the spatial autocorrelation.

The objective of this work is as follows. Suppose one has collected a set of image samples of surfaces, produced from some manufacturing process (the 16 samples in Figure 4.1 can be viewed as a subset of one such set). Further suppose that one suspects the process may not be stable, so that the stochastic nature of the surfaces systematically varies across the set of samples. That is,

the samples may not be a set of statistically equivalent realizations of the same underlying stochastic process. The goal is to characterize how the stochastic nature of the surfaces varies across the set of image samples, in a manner that is conducive to conveying an understanding of the physical nature of the variation. For example, the variation depicted in Figure 4.1 consists of two systematic variation patterns that represent a varying rate of spatial decay of the autocorrelation and a varying angle of orientation of the autocorrelation. Note that the two variation patterns occur simultaneously, that is, from image sample to image sample both the spatial decay level and orientation of the autocorrelation change concurrently. This makes it more difficult to recognize the physical meaning of each variation pattern, especially when the number of variation patterns increases. In this example, based only on the image samples and with no prior knowledge of the nature of the variation, we would like to empirically identify a two-dimensional manifold in which the two manifold coordinates represent a parameterization of the two systematic variation patterns. As an illustration, the coordinates of such a manifold parameterization, along with a visualization of how the parameters influence the stochastic nature of the surfaces, are shown in Figure 4.2 (we revisit this example in more detail in Section 4.4). Visualizing how the surfaces change as the manifold parameters are varied helps reveal the physical characteristic of each individual variation pattern, making it easier to identify the root cause problems for process improvement.

In the literature, there are several exploratory analysis methods to discover the nature of previously unidentified systematic variation. For example, Apley and Shi (2001), Apley and Lee (2003), Lee and Apley (2004) and Shan and Apley (2008) focused on linear variation patterns, whereas Apley and Zhang (2007) and Shi et al. (2016) focused on nonlinear variation patterns. However, all these methods are inapplicable for stochastic textured surface data due to their aforementioned stochastic nature. Chapters 2 and 3 developed monitoring approaches for stochastic textured surface data that seek to detect when some characteristic of their stochastic

behavior changes. In contrast, this work focuses on a more exploratory diagnostic objective of understanding the nature of the variation across a set of image samples, given that their stochastic behavior is not stable and varies across the set. This exploratory work can be viewed as a Phase I method, using the common Phase I versus Phase II distinction in the SPC literature (Human et al. 2010). In contrast, the works in Chapters 2 and 3 are Phase II methods, which focus on monitoring, as opposed to exploratory diagnostics. In this work, we derive two new pairwise dissimilarity measures for the image samples based on a novel application of the Kullback-Leibler divergence. Then, we use a form of manifold learning applied to the pairwise dissimilarities of the given set of samples. The learned manifold coordinates provide a parameterization of the variation existing in the set of samples, which can then be visualized as will be described later, akin to Figure 4.2.

This work also has broader applicability. In the context related to this work, which is understanding variation, our approach is applicable to a wide range of materials. This includes random heterogeneous materials, which are ubiquitous in science, engineering, and nature (Torquato 2002). Second, the derived dissimilarity measures can be used in other applications involving stochastic textured surface data in which some form of follow-up classification or clustering analysis is desired. Some examples include powder materials micrograph characterization (DeCost and Holm 2017), medical microscopy image classification (Jiang et al. 2015, Song et al. 2017), cancer tissue image clustering (Xu et al. 2014), and outlying mammalian cell image detection (Lou et al. 2012).

The remainder of this chapter is organized as follows. In Section 4.2, we derive and investigate the two new pairwise dissimilarity measures between image samples, which is a critical element of the approach. Section 4.3 describes how a form of manifold learning that takes pairwise dissimilarities as the input can be used for understanding the variation patterns existing in the image samples. Sections 4.4 and 4.5 demonstrate the approach and compare the different dissimilarity measures with a simulation example and a real textile example, respectively. Section

4.6 provides some further discussions on visualization for understanding variation, choice of parameters when computing the image sample dissimilarities, and choice of dissimilarity measures and manifold learning algorithms. Section 4.7 concludes the chapter.

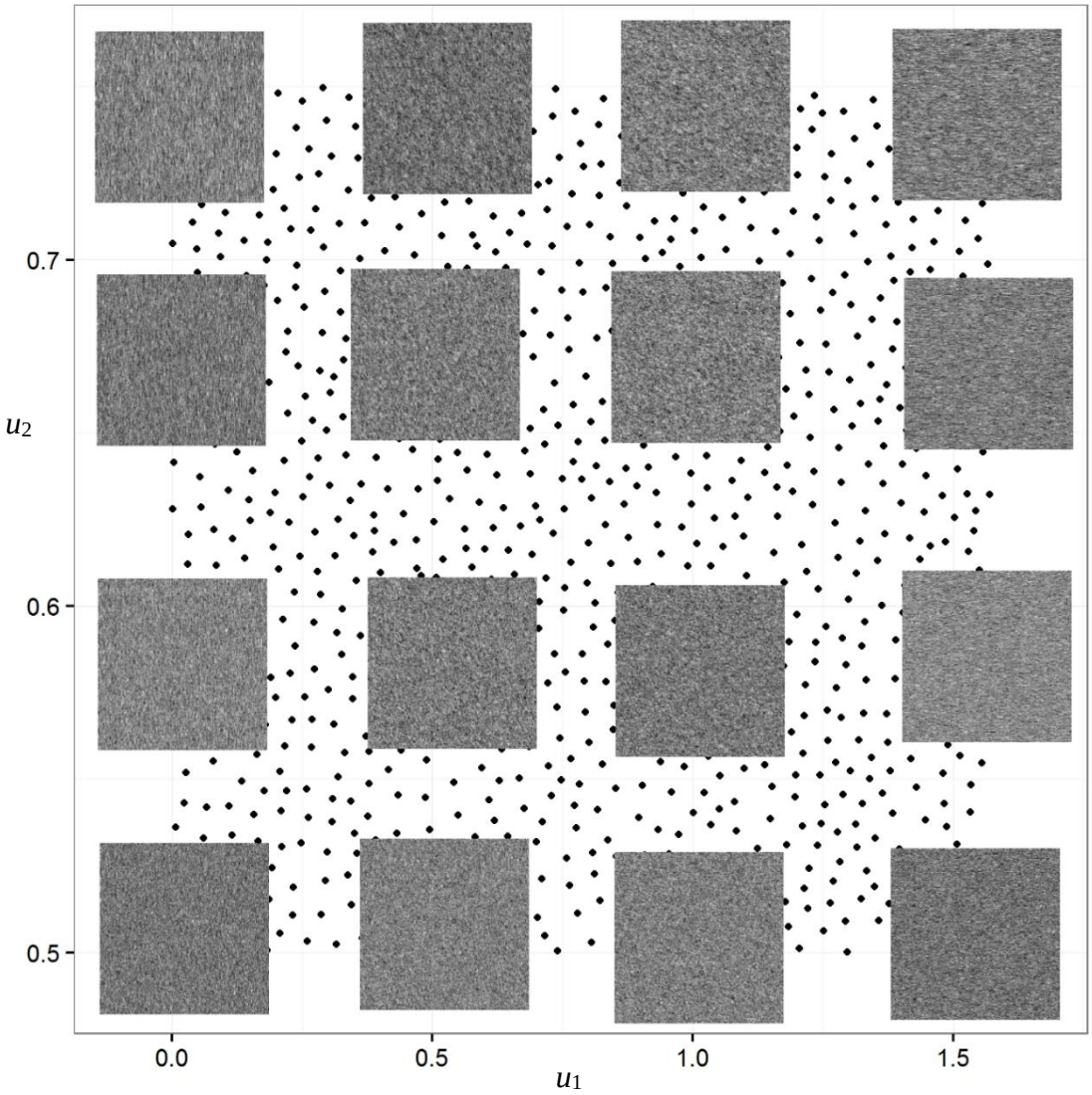


Figure 4.2. An illustrative manifold learning parameterization (u_1, u_2) of the variation across the image samples for the example depicted in Figure 4.1, along with superimposed image samples with stochastic nature corresponding to the specific value of (u_1, u_2) at the center of the image. As the learned parameter u_1 changes, the orientation angle changes. Likewise, as u_2 changes, the rate of decay of the spatial autocorrelation changes.

4.2 Deriving Pairwise Dissimilarity Measures for Stochastic Textured Surface Data

As discussed in Section 4.1, it is challenging to obtain pairwise dissimilarities between stochastic textured surface image samples due to their stochastic nature. In this section, we derive new dissimilarity measures for them based on Kullback-Leibler (KL) divergence principles. For univariate probability density functions (pdfs), the KL divergence is defined as follows. Let $p(x)$ and $q(x)$ be two univariate pdfs with $x \in \mathbb{R}$. The KL divergence of $q(x)$ from $p(x)$ measures the extent to which $q(x)$ differs from (or diverges from) $p(x)$ and is defined as:

$$D[p(x)||q(x)] = E_p\left[\log\frac{p(x)}{q(x)}\right],$$

where E_p denotes the expectation with respect to the distribution $p(\cdot)$. Similarly, for the multivariate case, let $p(\mathbf{x})$ and $q(\mathbf{x})$ be two multivariate joint pdfs, where the vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$. With a slight abuse of notation, we have used the same $p(\cdot)$ and $q(\cdot)$ to denote both the univariate and joint multivariate pdfs, since their distinction should be clear from their scalar or vector arguments. In this case, the KL divergence of $q(\mathbf{x})$ from $p(\mathbf{x})$ is defined as:

$$D[p(\mathbf{x})||q(\mathbf{x})] = E_p\left[\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\right].$$

Our proposed approach for measuring dissimilarities between image samples is based on the KL divergence concept derived using a particularly convenient representation of the joint pdfs of the image samples. Specifically, let $\mathbf{Y}_k = \left[y_{k,1}, y_{k,2}, \ldots, y_{k,M_k}\right]^T$ denote the vector of M_k pixel intensities in the kth image sample for $k = 1, 2, \ldots, N$ (it is assumed we have a set of N image samples, and the intermediate goal is to determine the dissimilarities between all pairs of image samples), stacked according to some raster scan order. In this chapter, we use a left-to-right and top-to-bottom raster scan order. Let $f_k(\mathbf{Y}_k)$ and $f_h(\mathbf{Y}_h)$ be the joint pdfs of data vectors $\mathbf{Y}_k$ and $\mathbf{Y}_h$, for image samples k and h, respectively, to be characterized as described below. Because a dissimilarity measure should be symmetric, we define the KL pairwise dissimilarity between the *k*th and *h*th image samples as

$$d_{\mathrm{KL}}(k,h)=\sqrt{\frac{1}{M_k}D[f_k(\mathbf{Y}_k)||f_h(\mathbf{Y}_k)]+\frac{1}{M_h}D[f_h(\mathbf{Y}_h)||f_k(\mathbf{Y}_h)]}, \tag{4.1}$$

where we have added the weights $1/M_k$ and $1/M_h$ to account for $\mathbf{Y}_k$ and $\mathbf{Y}_h$ possibly having different lengths. Multiplying, instead of summing, the two terms in (4.1) would also symmetrize the dissimilarity measure; however, we do not pursue this direction because later it will be clear that summing the two terms simplifies the derivation of the dissimilarity measure and its final form. We include the square root in (4.1) because it better mimics a 2-norm, considering the form (derived later) of the terms inside the radical sign. Also, including the square root improved the results in our experience.

The derivation of our KL dissimilarity measure is facilitated by the fact that the KL divergence of joint pdfs can be factored in terms of the KL divergence for univariate component distributions by the familiar chain rule (see Cover and Thomas 2006, for a more detailed derivation)

$$\begin{aligned}
D[p(\mathbf{x})||q(\mathbf{x})] &= E_p\left[\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\right]=E_p\log\left[\frac{p(x_1)p(x_2|x_1)\ldots p(x_n|x_{n-1},x_{n-2},\ldots,x_1)}{q(x_1)q(x_2|x_1)\ldots q(x_n|x_{n-1},x_{n-2},\ldots,x_1)}\right]\\
&= E_p\left[\log\frac{p(x_1)}{q(x_1)}+\log\frac{p(x_2|x_1)}{q(x_2|x_1)}+\cdots+\log\frac{p(x_n|x_{n-1},x_{n-2},\ldots,x_1)}{q(x_n|x_{n-1},x_{n-2},\ldots,x_1)}\right]\\
&= E_p\left[\log\frac{p(x_1)}{q(x_1)}\right]+E_p\left[\log\frac{p(x_2|x_1)}{q(x_2|x_1)}\right]+\cdots+E_p\left[\log\frac{p(x_n|x_{n-1},x_{n-2},\ldots,x_1)}{q(x_n|x_{n-1},x_{n-2},\ldots,x_1)}\right]\\
&= D[p(x_1)||q(x_1)]+D[p(x_2|x_1)||q(x_2|x_1)]+\cdots\\
&\qquad +D[p(x_n|x_{n-1},x_{n-2},\ldots,x_1)||q(x_n|x_{n-1},x_{n-2},\ldots,x_1)].
\end{aligned}$$

Using the chain rule, we have:

$$\begin{aligned}
D[f_k(\mathbf{Y}_k)||f_h(\mathbf{Y}_k)] &= D\left[f_k(y_{k,1})||f_h(y_{k,1})\right]+D\left[f_k(y_{k,2}|y_{k,1})||f_h(y_{k,2}|y_{k,1})\right]+\cdots\\
&\qquad +D\left[f_k(y_{k,n}|y_{k,n-1},y_{k,n-2},\ldots,y_{k,1})||f_h(y_{k,n}|y_{k,n-1},y_{k,n-2},\ldots,y_{k,1})\right]\\
&= \textstyle\sum_{i=1}^{M_k} D\left[f_k\left(y_{k,i}\middle|\mathbf{Y}_k^{(i)}\right)||f_h\left(y_{k,i}\middle|\mathbf{Y}_k^{(i)}\right)\right],
\end{aligned} \tag{4.2}$$

where we have defined $\mathbf{Y}_k^{(i)} = \{y_{k,m}: m < i\}$ to be the set of pixel intensities prior to pixel i in image k. Again, with some abuse of notation, we have denoted the joint distribution and univariate conditional distributions for image k by the same quantity $f_k(\cdot)$, since it should be clear from their arguments which it represents. Also notice that we did not add a second, pixel subscript i to the conditional distributions $f_k\left(y_{k,i}\middle|\mathbf{Y}_k^{(i)}\right)$ in (4.2), because of the stationarity assumption discussed below.

From (4.2), it follows that

$$D[f_h(\mathbf{Y}_h)||f_k(\mathbf{Y}_h)] = \sum_{i=1}^{M_h} D\left[f_h\left(y_{h,i}\middle|\mathbf{Y}_h^{(i)}\right)||f_k\left(y_{h,i}\middle|\mathbf{Y}_h^{(i)}\right)\right]. \tag{4.3}$$

Plugging (4.2) and (4.3) to (4.1), we have:

$$d_{\mathrm{KL}}(k,h) = \sqrt{\frac{\sum_{i=1}^{M_k} D\left[f_k\left(y_{k,i}\middle|\mathbf{Y}_k^{(i)}\right)||f_h\left(y_{k,i}\middle|\mathbf{Y}_k^{(i)}\right)\right]}{M_k} + \frac{\sum_{i=1}^{M_h} D\left[f_h\left(y_{h,i}\middle|\mathbf{Y}_h^{(i)}\right)||f_k\left(y_{h,i}\middle|\mathbf{Y}_h^{(i)}\right)\right]}{M_h}}. \tag{4.4}$$

Using (4.4), computing the KL dissimilarity between the *k*th and *h*th image samples boils down to estimating the conditional distributions $f_j\left(y_{j,i}\middle|\mathbf{Y}_j^{(i)}\right)$ for each pixel *i* in the image samples $j = k, h$. Below, we describe how these conditional distributions can be approximated and estimated quite tractably using two Markov random field assumptions.

In texture synthesis problems (e.g., see Efros and Leung 1999, Levina and Bickel 2006), it is often assumed that there exists some moving window neighborhood $\mathbf{y}_j^{(i)} \subseteq \mathbf{Y}_j^{(i)}$ of fixed size such that the Markov locality property $f_j\left(y_{j,i}\middle|\mathbf{Y}_j^{(i)}\right) \approx f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right)$ holds. In addition, $f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right)$ is assumed to be stationary, i.e., $f_j\left(y_{j,i} = y\middle|\mathbf{y}_j^{(i)} = \mathbf{y}\right)$, as a function of *y* and **y**, is independent of the pixel location *i*. For each pixel, we use a rectangular neighborhood of the form shown as the black regions in Figure 4.3(a) for pixels $i = 103$ and $i = 248$, extending to the right, left, and above pixel i. Let *l* denote the size of the Markov neighborhood in an image, i.e., the number of pixels to the left of, right of, and above pixel *i*. For $l = 2$, the neighborhoods corresponding to pixels y_{103} and y_{248} for the 50×50 image in Figure 4.3 are $\mathbf{y}^{(103)} = \{y_1, \dots, y_5, y_{51}, \dots, y_{55}, y_{101}, y_{102}\}$ and

$\mathbf{y}^{(248)} = \{y_{146}, \ldots, y_{150}, y_{196}, \ldots, y_{200}, y_{246}, y_{247}\}$, respectively, as shown on the right side of Figure 4.3(a). If $l = 2$ is large enough for the Markov locality assumption to hold, we have that $f\left(y_{103}\middle|\mathbf{Y}^{(103)}\right) \approx f\left(y_{103}\middle|\mathbf{y}^{(103)}\right)$ and $f\left(y_{248}\middle|\mathbf{Y}^{(248)}\right) \approx f\left(y_{248}\middle|\mathbf{y}^{(248)}\right)$. If in addition the stationarity assumption holds, then $f\left(y_{103} = y\middle|\mathbf{y}^{(103)} = \mathbf{y}\right) = f\left(y_{248} = y\middle|\mathbf{y}^{(248)} = \mathbf{y}\right)$.

In this chapter, we also assume that these two Markov random field assumptions hold, and hence from (4.4):

$$d_{\mathrm{KL}}(k,h) \approx \sqrt{\frac{\sum_{i=1}^{M_k} D\left[f_k\left(y_{k,i}\middle|\mathbf{y}_k^{(i)}\right)||f_h\left(y_{k,i}\middle|\mathbf{y}_k^{(i)}\right)\right]}{M_k} + \frac{\sum_{i=1}^{M_h} D\left[f_h\left(y_{h,i}\middle|\mathbf{y}_h^{(i)}\right)||f_k\left(y_{h,i}\middle|\mathbf{y}_h^{(i)}\right)\right]}{M_h}}$$

(by the locality assumption)

$$= \sqrt{E_{f_k}\left[\log\frac{f_k\left(y_{k,i}\middle|\mathbf{y}_k^{(i)}\right)}{f_h\left(y_{k,i}\middle|\mathbf{y}_k^{(i)}\right)}\right] + E_{f_h}\left[\log\frac{f_h\left(y_{h,i}\middle|\mathbf{y}_h^{(i)}\right)}{f_k\left(y_{h,i}\middle|\mathbf{y}_h^{(i)}\right)}\right]} \tag{4.5}$$

(by the stationarity assumption and definition of the KL divergence).

Although there are four conditional distribution terms in (4.5), there are really only two distinct conditional distribution functions $f_j(\cdot \mid \cdot), j = k, h$. Therefore, calculating the KL dissimilarity between the *k*th and *h*th image samples simplifies to estimating two stationary conditional distribution functions $f_j(\cdot \mid \cdot), j = k, h$, which is accomplished as follows.

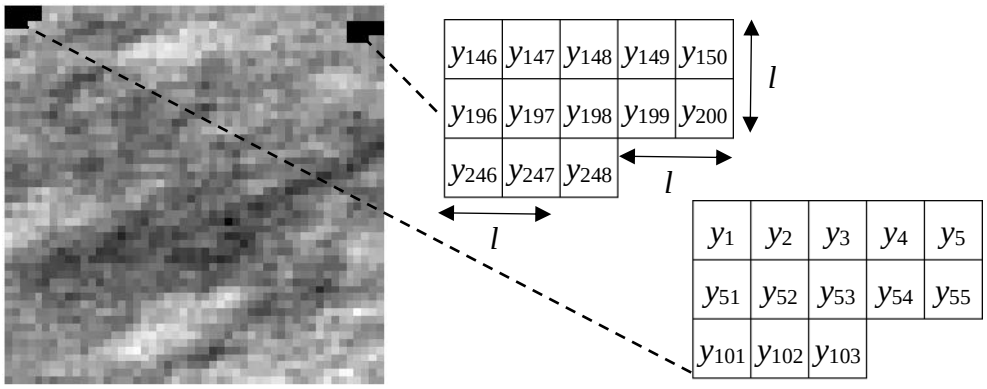
For online monitoring of greyscale image samples that are finely discretized (enough to be treated as continuous), based on the stationarity assumption, Chapter 2 adapted the approach of Bostanabad et al. (2016) to learn $f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right)$ via fitting the following supervised learning (aka nonlinear regression) model[1] to a set of training data constructed from image sample *j* such that the *i*th row in this data set consists of $\left\{y_{j,i}, \mathbf{y}_j^{(i)}\right\}, i = 1, 2, \ldots, M_j$:

$$y_{j,i} = g_j\left(\mathbf{y}_j^{(i)}\right) + \varepsilon_{j,i}, \tag{4.6}$$

[1] The method of Bostanabad et al. (2016) is for (two-phase) binary microstructure image characterization and reconstruction. In their context, the conditional distribution is categorical, and a classifier is used instead of a nonlinear regression model to learn it.

where $g_j\left(\mathbf{y}_j^{(i)}\right)$ is the mean of the conditional distribution $f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right)$, and $\varepsilon_{j,i}$ is a zero-mean random error that is independent of $\mathbf{y}_j^{(i)}$. The function $g_j(\cdot)$ is essentially the supervised learning model for predicting a pixel value $y_{j,i}$, given its neighboring pixel values $\mathbf{y}_j^{(i)}$. More specifically, in the *i*th row of the training data, the first element contains the intensity $y_{j,i}$ of the *i*th pixel and the other elements contain the intensities $\mathbf{y}_j^{(i)}$ of the neighbors of the *i*th pixel (see Figure 4.3(b) for an illustrative example). Chapter 2 used regression trees as the supervised learning model to estimate $\hat{g}_j(\cdot)$ from the training data.

(a)



(b)

y_i	$\mathbf{y}^{(i)}$											
⋮												
y_{103} ← 103rd row	y_1	y_2	y_3	y_4	y_5	y_{51}	y_{52}	y_{53}	y_{54}	y_{55}	y_{101}	y_{102}
⋮												
y_{248} ← 248th row	y_{146}	y_{147}	y_{148}	y_{149}	y_{150}	y_{196}	y_{197}	y_{198}	y_{199}	y_{200}	y_{246}	y_{247}
⋮												

Figure 4.3. (a) Illustration of neighborhoods of size $l = 2$ in an image of size 50×50 pixels, using the left-to-right and top-to-bottom raster scan order (i.e., the intensities of the top left, top right, and bottom right pixels are y_1, y_{50}, and y_{2500}, respectively). The two black regions on the image highlight pixels y_{103} and y_{248} and their corresponding neighborhoods. (b) Illustration of the 103rd and 248th rows of the training data corresponding to the image in Panel (a).

For tractability, we assume additionally that $\varepsilon_{j,i}$ follows an independent normal distribution

with variance σ_j^2, then $f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right) \equiv N(g_j\left(\mathbf{y}_j^{(i)}\right), \sigma_j^2)$, i.e., the normal pdf with mean $g_j\left(\mathbf{y}_j^{(i)}\right)$ and variance σ_j^2. In Remark 4, we discuss why the normal distribution is still an appealing choice even if $f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right)$ is not normal. Note that σ_j^2 can be estimated by

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{M_j}\left[y_{j,i} - \hat{g}_j\left(\mathbf{y}_j^{(i)}\right)\right]^2}{M_j}. \tag{4.7}$$

From the preceding, the quantities involved in (4.5) become

$$E_{f_k}\left[\log\frac{f_k\left(y_{k,i}\middle|\mathbf{y}_k^{(i)}\right)}{f_h\left(y_{k,i}\middle|\mathbf{y}_k^{(i)}\right)}\right] \approx E_{f_k}\left[\log\frac{\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left\{-\frac{\left(y_{k,i}-g_k\left(\mathbf{y}_k^{(i)}\right)\right)^2}{2\sigma_k^2}\right\}}{\frac{1}{\sqrt{2\pi}\sigma_h}\exp\left\{-\frac{\left(y_{k,i}-g_h\left(\mathbf{y}_k^{(i)}\right)\right)^2}{2\sigma_h^2}\right\}}\right]$$

$$= E_{f_k}\left[\log\frac{\sigma_h}{\sigma_k} - \frac{\left(y_{k,i}-g_k\left(\mathbf{y}_k^{(i)}\right)\right)^2}{2\sigma_k^2} + \frac{\left(y_{k,i}-g_h\left(\mathbf{y}_k^{(i)}\right)\right)^2}{2\sigma_h^2}\right] = \log\frac{\sigma_h}{\sigma_k} - \frac{1}{2} + \frac{E_{f_k}\left[\left(y_{k,i}-g_h\left(\mathbf{y}_k^{(i)}\right)\right)^2\right]}{2\sigma_h^2}$$

$$= \log\frac{\sigma_h}{\sigma_k} - \frac{1}{2} + \frac{E_{f_k}\left[\left(y_{k,i}-g_k\left(\mathbf{y}_k^{(i)}\right)+g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right)^2\right]}{2\sigma_h^2}$$

$$= \log\frac{\sigma_h}{\sigma_k} - \frac{1}{2} + \frac{E_{f_k}\left[\left(y_{k,i}-g_k\left(\mathbf{y}_k^{(i)}\right)\right)^2\right]+2E_{f_k}\left[\left(y_{k,i}-g_k\left(\mathbf{y}_k^{(i)}\right)\right)\left(g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right)\right]+E_{f_k}\left[\left(g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right)^2\right]}{2\sigma_h^2}$$

$$= \log\frac{\sigma_h}{\sigma_k} - \frac{1}{2} + \frac{\sigma_k^2}{2\sigma_h^2} + \frac{E_{f_k}\left[\left(g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right)^2\right]}{2\sigma_h^2}$$

$$\approx \log\frac{\sigma_h}{\sigma_k} - \frac{1}{2} + \frac{\sigma_k^2}{2\sigma_h^2} + \frac{\sum_{i=1}^{M_k}\left[g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right]^2}{2M_k\sigma_h^2}. \tag{4.8}$$

Similarly, we have:

$$E_{f_h}\left[\log\frac{f_h\left(y_{h,i}\middle|\mathbf{y}_h^{(i)}\right)}{f_k\left(y_{h,i}\middle|\mathbf{y}_h^{(i)}\right)}\right] \approx \log\frac{\sigma_k}{\sigma_h} - \frac{1}{2} + \frac{\sigma_h^2}{2\sigma_k^2} + \frac{\sum_{i=1}^{M_h}\left[g_h\left(\mathbf{y}_h^{(i)}\right)-g_k\left(\mathbf{y}_h^{(i)}\right)\right]^2}{2M_h\sigma_k^2}. \tag{4.9}$$

Plugging (4.8) and (4.9) in (4.5), the KL dissimilarity between the *k*th and *h*th image samples is:

$$d_{\mathrm{KL}}(k,h) \approx \sqrt{-1+\frac{1}{2}\left(\frac{\sigma_k^2}{\sigma_h^2}+\frac{\sigma_h^2}{\sigma_k^2}\right)+\frac{\sum_{i=1}^{M_k}\left[g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right]^2}{2M_k\sigma_h^2}+\frac{\sum_{i=1}^{M_h}\left[g_h\left(\mathbf{y}_h^{(i)}\right)-g_k\left(\mathbf{y}_h^{(i)}\right)\right]^2}{2M_h\sigma_k^2}}, \quad (4.10)$$

where $\{g_k(\cdot), \sigma_k^2\}$ and $\{g_h(\cdot), \sigma_h^2\}$ are estimated using the supervised learning approach.

Remark 1: The argument within the square root in (4.10) can be easily shown to be non-negative using the Cauchy inequality, and hence the KL dissimilarity is always real.

Remark 2: If we assume a common error variance $\sigma_k^2 = \sigma^2$ for each image $k = 1, 2, \ldots, N$, then (4.10) reduces to:

$$d_{\mathrm{KL}}(k,h) = \sqrt{\frac{\sum_{i=1}^{M_k}\left[g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right]^2}{2M_k\sigma^2}+\frac{\sum_{i=1}^{M_h}\left[g_h\left(\mathbf{y}_h^{(i)}\right)-g_k\left(\mathbf{y}_h^{(i)}\right)\right]^2}{2M_h\sigma^2}}.$$

Under the common error variance assumption, and omitting the common factor $2\sigma^2$, we define the approximate KL (AKL) dissimilarity between the *k*th and *h*th image samples as:

$$d_{\mathrm{AKL}}(k,h) = \sqrt{\frac{\sum_{i=1}^{M_k}\left[g_k\left(\mathbf{y}_k^{(i)}\right)-g_h\left(\mathbf{y}_k^{(i)}\right)\right]^2}{M_k}+\frac{\sum_{i=1}^{M_h}\left[g_h\left(\mathbf{y}_h^{(i)}\right)-g_k\left(\mathbf{y}_h^{(i)}\right)\right]^2}{M_h}}. \quad (4.11)$$

Remark 3: Notice that when $M_k = M_h$, the AKL dissimilarity is equivalent to the Euclidean distance between the vectors of predictions from the two supervised learning models applied to the *k*th and *h*th image samples, i.e. the vectors $\left[\hat{g}_k\left(\mathbf{y}_k^{(1)}\right) \ldots \hat{g}_k\left(\mathbf{y}_k^{(M_k)}\right), \hat{g}_k\left(\mathbf{y}_h^{(1)}\right) \ldots \hat{g}_k\left(\mathbf{y}_h^{(M_h)}\right)\right]^T$ and $\left[\hat{g}_h\left(\mathbf{y}_k^{(1)}\right) \ldots \hat{g}_h\left(\mathbf{y}_k^{(M_k)}\right), \hat{g}_h\left(\mathbf{y}_h^{(1)}\right) \ldots \hat{g}_h\left(\mathbf{y}_h^{(M_h)}\right)\right]^T$.

Remark 4: Even if the prediction errors $\varepsilon_{j,i}$ are not truly iid and normally distributed, the iid normality assumption for $f_j\left(y_{j,i}\middle|\mathbf{y}_j^{(i)}\right)$ can be viewed as a convenient means to an attractive end. That is, even if the errors are not normal, the final form of the KL dissimilarity in (4.10) under the normal assumption is conceptually appealing. When the *k*th and *h*th image samples are similar, we expect $g_k(\cdot) \approx g_h(\cdot)$ and $\sigma_k \approx \sigma_h$, in which case $d_{\mathrm{KL}}(k,h) \approx 0$. In contrast, when the *k*th and *h*th image samples are different, we expect $g_k(\cdot) \neq g_h(\cdot)$ and/or $\sigma_k \neq \sigma_h$, in which case $d_{\mathrm{KL}}(k,h) > 0$ regardless of the true error distribution. The same argument applies to the AKL dissimilarity.

4.3 Identifying and Visualizing the Nature of Variation in Stochastic Textured Surface Data

Given a set of *N* image samples, the overarching goal of this work is to discover the nature of systematic variation in the stochastic nature of the images across the set of image samples, in a manner that helps understand the physical meaning of individual variation patterns. This in turn may help identify the root causes of the variation more easily. In this section, we present our approach and algorithm for accomplishing this, based on the dissimilarity measures developed in Section 4.2. In particular, we use a form of manifold learning with dissimilarity data to identify and parameterize the variation patterns, followed by a graphical visualization of the identified manifold learning parameters. In the following paragraphs, we first briefly review certain manifold learning concepts before describing our approach.

Consider a set of *N* data points $\mathbf{z}_j = \left[z_{j,1}, z_{j,2}, \dots, z_{j,n}\right]^T \in \mathbb{R}^n, j = 1, 2, \dots, N$ that lies on an unknown *r*-dimensional manifold embedded in the *n*-dimensional space, where $r < n$. For illustration, Figure 4.4(a) shows data points in a space of dimension $n = 3$ that lie on a manifold of dimension $r = 2$ embedded in the 3-dimensional space. Given only $\{\mathbf{z}_j : j = 1, 2, \dots, N\}$, manifold learning aims to estimate the low *r*-dimensional manifold coordinates of $\{\mathbf{z}_j : j = 1, 2, \dots, N\}$. In other words, manifold learning techniques find the manifold coordinates $\mathbf{u}_j = \left[u_{j,1}, u_{j,2}, \dots, u_{j,r}\right]^T \in \mathbb{R}^r$ such that $\mathbf{z}_j \approx \mathbf{h}(\mathbf{u}_j) = \left[h_1(\mathbf{u}_j), h_2(\mathbf{u}_j), \dots, h_n(\mathbf{u}_j)\right]^T$ for $j = 1, 2, \dots, N$ and for some implicit mapping $\mathbf{h}(\cdot): \mathbb{R}^r \rightarrow \mathbb{R}^n$. Figure 4.4(b) illustrates this by plotting the two-dimensional manifold coordinates $\mathbf{u}_j \in \mathbb{R}^2$ of the data points in Figure 4.4(a). The two-dimensional manifold coordinates can be viewed as an unfolded version of the nonlinear manifold embedded in the original *n*-dimensional space, and the goal of the manifold learning algorithm is to estimate the manifold coordinates. Note that manifold learning only produces the manifold coordinates and does not produce the implicit function $\mathbf{h}(\cdot)$.

If the *n*-dimensional data points $\{\mathbf{z}_j : j = 1, 2, \dots, N\}$ are not available, and only their pairwise dissimilarities $\{d(k, h): 1 \leq k < h \leq N\}$ are available, many standard manifold learning methods

cannot be used. Instead, a form of manifold learning such as the multidimensional scaling method (MDS) of Torgerson (1952) or the isometric feature mapping method (ISOMAP) of Tenenbaum et al. (2000), which can work with only the dissimilarity data, can be used to estimate the manifold coordinates. For example, MDS finds manifold coordinates $\mathbf{u}_k$ and $\mathbf{u}_h$ such that the Euclidean distance between $\mathbf{u}_k$ and $\mathbf{u}_h$ is approximately $d(k,h)$ for all $1 \leq k < h \leq N$ via an eigen-decomposition of a doubly centered $N \times N$ dissimilarity matrix. ISOMAP also solves an eigen-problem to produce the manifold coordinates; however, instead of preserving the global structure of the manifold as in MDS, ISOMAP preserves the neighborhood structure on the manifold. As such, ISOMAP can unfold highly nonlinear manifold better than MDS. Readers are referred to Izenman (2013) for details of these algorithms.

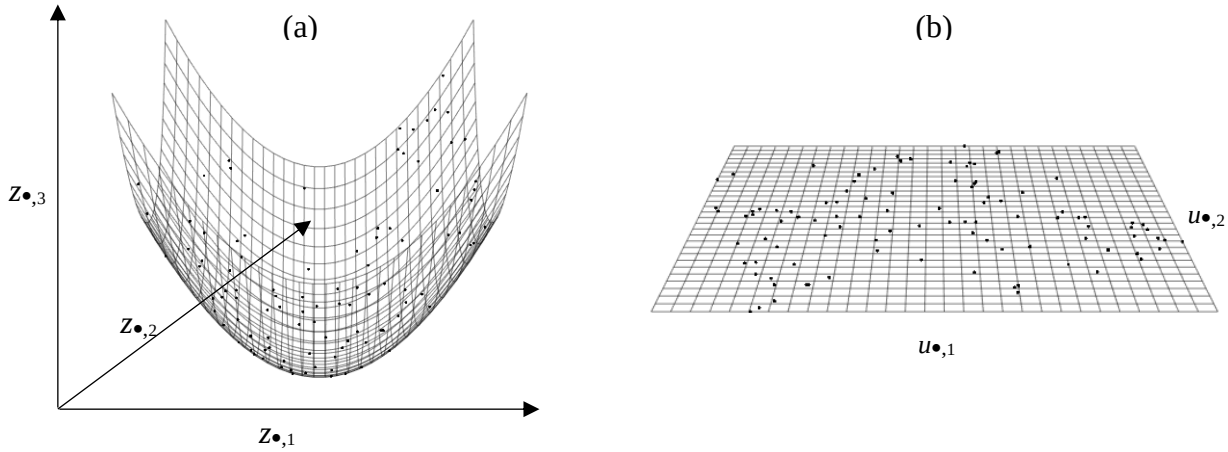


Figure 4.4. Illustration of general manifold learning: Panel (a) plots a set of 3-dimensional data points lying on a two-dimensional manifold embedded in the 3-dimensional space. Manifold learning unfolds the manifold and obtains the two-dimensional manifold coordinates of the data points in Panel (a).

As discussed in Section 4.1, the image samples in our problem cannot be compared pixel-by-pixel (nor can they be transformed for such purposes), and thus, we cannot treat each image sample as an input vector $\mathbf{z}$ to be used in manifold learning algorithms that require the original data. However, using our pairwise dissimilarity measures that we developed in Section 4.2, we can apply one of the forms of manifold learning that only require dissimilarities.

As mentioned earlier, our goal is to understand the physical meaning of any variation in the stochastic nature of the images across the set of image samples. Simply inspecting the original image samples may not reveal the nature of the patterns, especially when multiple patterns are mixed together and buried in high noise levels. This is the motivation for learning the manifold coordinates of the image samples, because they provide a low-dimensional parameterization of the set of systematic variation patterns present in the data, in a manner that inherently filters out noise. By inspecting the image samples along some particular lines or curves in the learned manifold coordinate space, the individual variation patterns can be revealed. We discuss and illustrate this step with the examples in Sections 4.4 and 4.5.

The steps of the entire procedure for identifying and visualizing the variation patterns are summarized as follows, with certain details deferred until Sections 4.4 and 4.5:

Step 1: Compute the $N \times N$ symmetric dissimilarity matrix $\mathbf{D} = \{d(k,h): 1 \leq k,h \leq N\}$ for the given set of N image samples by performing the following steps:

- Step 1(a): Fit the regression model (4.6) for each image sample to obtain $\{\hat{g}_j(\cdot) : j = 1, 2, \ldots, N\}$. Section 4.6.2 will discuss how to choose the parameters needed in this step via CV.
- Step 1(b): Compute $\{\hat{\sigma}_j^2 : j = 1, 2, \ldots, N\}$ using (4.7).
- Step 1(c): Compute the KL or AKL dissimilarities $\{d(k,h): 1 \leq k < h \leq N\}$, using $\{\hat{g}_j(\cdot), \hat{\sigma}_j^2 : j = 1, 2, \ldots, N\}$ obtained in Steps 1(a, b) and the KL or AKL dissimilarity measures. Section 4.6.3 discusses the choice of dissimilarity measure.

Step 2: Find the manifold coordinates of the N image samples by applying a manifold learning algorithm that takes pairwise dissimilarities as the input (e.g., MDS or ISOMAP) to **D**. When using an eigen-decomposition-based algorithm such as MDS or ISOMAP, we recommend using the number of dominant eigenvalues as the manifold dimension (if the manifold dimension is unknown). Note that the dimension of the manifold is the number of distinct variation patterns that are present. Section 4.6.3 discusses the choice of the manifold learning algorithm.

Step 3: Visualize the nature of the variation patterns existing in the given set of *N* image samples by inspecting the image samples along some particular lines or curves in the learned manifold coordinate space, as illustrated in the examples that follow (see also Section 4.6.1 for further discussion of this step).

4.4 Simulation Example

We first illustrate our approach using simulated image samples for which we know what the true nature of the variation patterns is. The next section illustrates with a real data example. The pixel intensities of the image samples in the simulation example were generated from the spatial autoregressive model

$$y(i_1, i_2) = \phi_1 y(i_1 - 1, i_2) + \phi_2 y(i_1, i_2 - 1) + \varepsilon(i_1, i_2), \tag{4.12}$$

where $y(i_1, i_2)$ denotes the image intensity at pixel location (i_1, i_2) with i_1 and i_2 the horizontal and vertical indices, respectively, and ε is a zero-mean Gaussian white noise process over the two-dimensional image. In this model, the transformed parameters $A = \phi_1 + \phi_2$ and $\gamma = \tan^{-1}(\phi_2/\phi_1)$ represent the spatial decay level and the orientation angle of the autocorrelation in the image samples, respectively. To create variation in the stochastic nature of the images across a set of image samples, we generate a different value of A and γ for each image. Hence, the rate of decay of the spatial autocorrelation in the simulated image samples varies as A varies from image to image. Likewise, the orientation angle of the spatial autocorrelation in the simulated image samples varies as γ varies from image to image. See Song and Kang (2018) for analogous changes in the parameters of ARMA-GARCH models for univariate time-series data.

We used Latin hypercube sampling (LHS) to generate $N = 200$ stratified random values of $A \in [0.5, 0.8]$ and $\gamma \in \left[\pi/8, 3\pi/8\right]$, and then computed the corresponding $\{\phi_1, \phi_2\}$ to simulate 200 image samples using (4.12). Figure 4.5(a) and Figure 4.5(b) plot the true values of $\{A, \gamma\}$ and the corresponding $\{\phi_1, \phi_2\}$ for these 200 image samples, respectively. The numbers in both Figure

4.5(a) and Figure 4.5(b) represent the image indices. All the simulated image samples have the same size of 250×250 pixels. Note that the MRF assumptions hold for the image samples in this example, by construction. In practice, with real images obtained by imaging methods in which lighting conditions vary from image to image, it is helpful to standardize all image samples by subtracting from each pixel the average intensity for all pixels in that image and then dividing by the intensity standard deviation for all pixels in the image.

Step 1 of the algorithm computes the dissimilarity matrix for the simulated set of image samples. First, in Step 1(a), we fitted a regression model for each simulated image sample $j = 1, 2, \ldots, N$. Although (4.12) is a linear model, we treated this knowledge as unknown and used regression trees to estimate $\{\hat{g}_j(\cdot): j = 1, 2, \ldots, N\}$ because they can model general regression functions and are relatively fast to fit. We used the **rpart** R package (Therneau and Atkinson 2018) to fit the regression trees, and this package requires specification of a complexity parameter (denoted by *cp*) that determines the size of the fitted tree. Here, we used the same *cp* = 0.0005 value for the trees fitted to each image (this value was chosen using the CV procedure described in Section 6.2). We also used CV to select the neighborhood size *l* when constructing the training data to fit $\{\hat{g}_j(\cdot): j = 1, 2, \ldots, N\}$ (see also Section 6.2 for details of the CV procedure to choose *l*). In this example, the chosen value was *l* = 1, which agreed with the true lag-1 model of Eq. (4.12). In Step 1(b), the AKL dissimilarity was computed for each pair of image samples using the fitted $\{\hat{g}_j(\cdot): j = 1, 2, \ldots, N\}$ in Step 1(a).

The outcome of Step 1 is a dissimilarity matrix, and we used it as the input for the MDS manifold learning algorithm in Step 2 to estimate the manifold coordinates of the *N* = 200 image samples. To select the number of dimensions to retain in the estimated manifold, we inspected the eigenvalues obtained when applying MDS to the dissimilarity matrix. The solid curve marked with "*" symbols in Figure 4.6 shows the ten largest eigenvalues, indicating there are two dominant eigenvalues in this example. This correctly indicated that there were two major systematic

variation patterns existing in the image samples (represented by image-to-image variation in the two parameters A and γ).

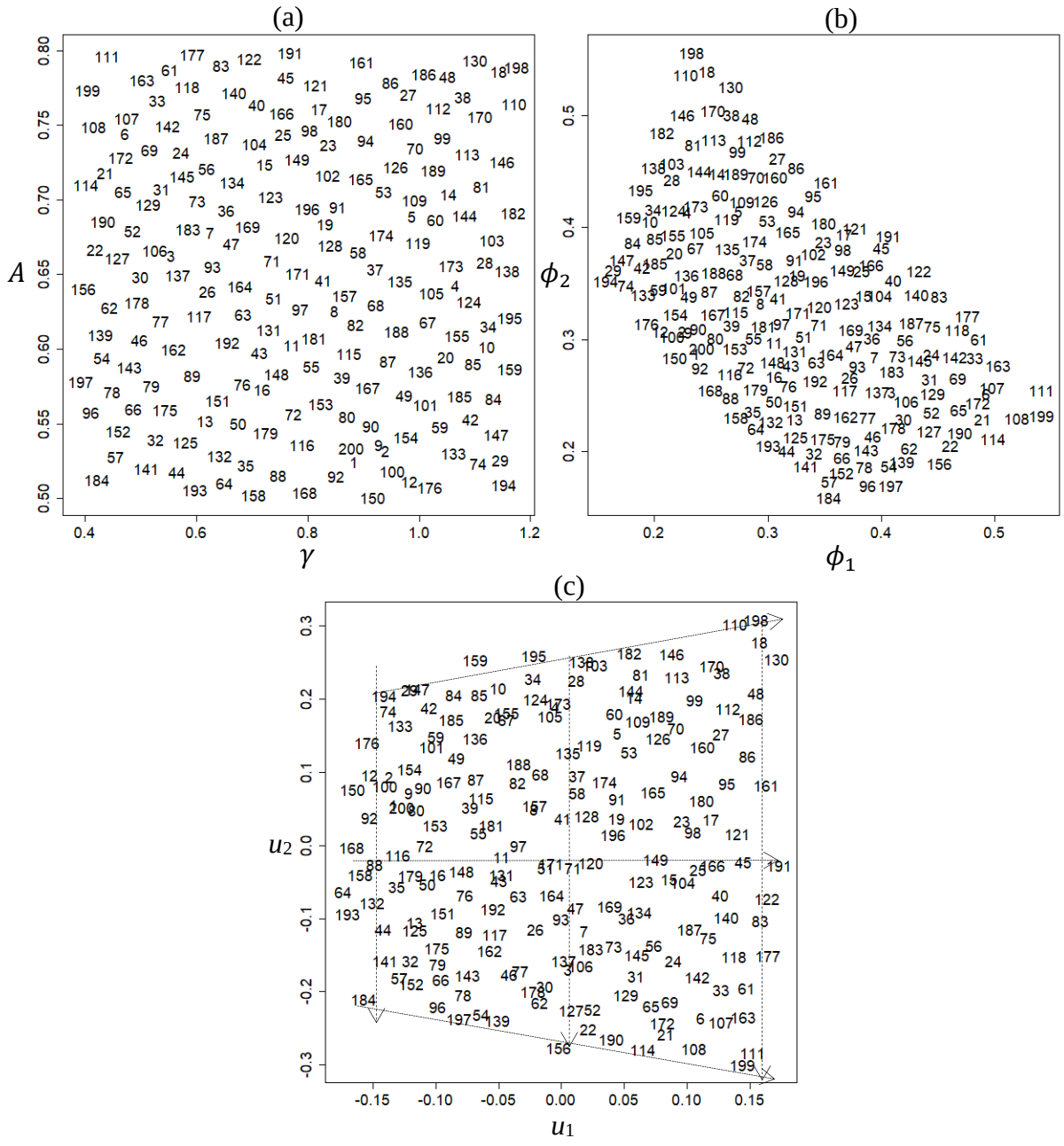


Figure 4.5. Some results in the simulation example: (a) Generated parameters $\{A, \gamma\}$ for the 200 image samples which represent two underlying variation patterns in the stochastic nature of the image samples and (b) their corresponding parameters $\{\phi_1, \phi_2\}$; (c) the estimated manifold coordinates from MDS applied to the AKL dissimilarity for the image samples generated by these parameters. The numbers in each panel are the image indices.

Remark 5: The dashed curve marked with circles in Figure 4.6 shows the ten largest eigenvalues of a simulation example without any systematic variation pattern, i.e., all images are simulated using the same values of $\{\phi_1, \phi_2\}$. In this case, the curve is relatively smooth with no obvious dominant eigenvalues, as opposed to the case with two systematic variation patterns.

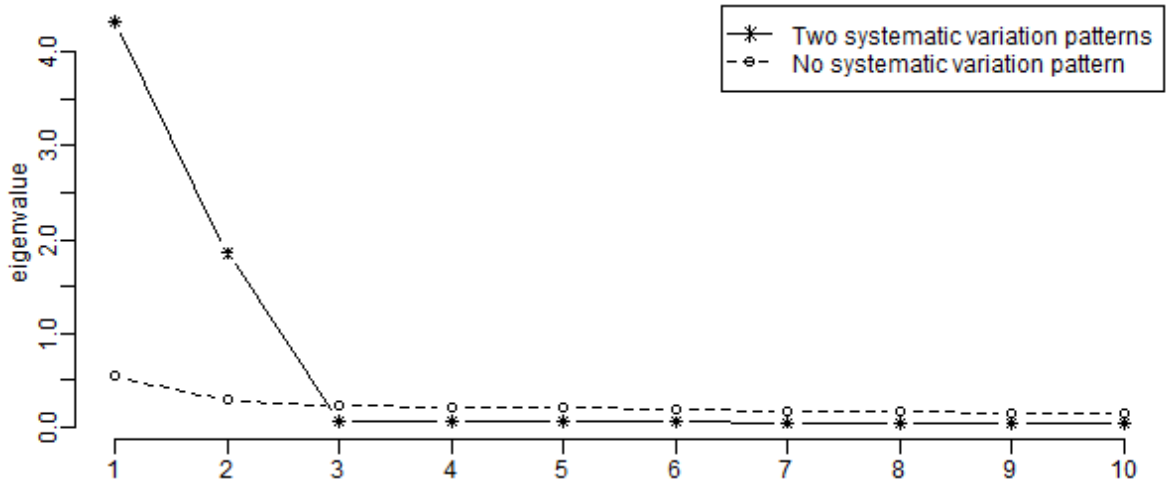


Figure 4.6. Scree plot of the 10 largest eigenvalues obtained from MDS applied to the AKL dissimilarity matrix for the simulation example, which is used to estimate the number (two) of variation patterns. For comparison, the curve marked with open circles shows the 10 largest eigenvalues for an example in which all images were generated using the same model, so that there were truly no variation patterns.

Figure 4.5(c) shows the actual estimated coordinates in Step 2, which interestingly resembles quite well the ground-truths of $\{A, \gamma\}$ (imagine Figure 4.5(a) being flipped such that the top left corner goes to the bottom right corner) and $\{\phi_1, \phi_2\}$ (imagine Figure 4.5(b) being rotated clockwise by 45 degrees and scaled). We will show quantitative measures for these resemblances below. The fact that MDS accurately estimated the ground-truth of the parameters used to generate the image samples means that the AKL dissimilarity captured very well the dissimilarities between the image samples. We emphasize that these results were obtained without incorporating any prior knowledge of the nature of the variation occurring in the samples.

In Step 3, we visualized the individual variation patterns existing in the image samples by

taking advantage of the low-dimensional parameterization of these variation patterns via the estimated manifold coordinates, as follows. First, we investigated the image samples that fell close to a trajectory of points on each of the six arrows on the plot of the estimated manifold coordinates of the image samples shown in Figure 4.5(c). Each arrow represents a path (linear, in this case) in the manifold coordinate space, and inspection of a set of image samples that fall close to the trajectory of points along the arrow shows how the stochastic nature of the image samples varies as the manifold coordinate parameters vary along that path. The directions of the arrows indicate the ordering of the image samples along the arrows that we investigated.

Due to space limit, Figure 4.7 shows only nine image samples that correspond to the intersections of the six arrows. To improve visibility, these image samples are magnified and cropped versions of the original ones (which were of the size shown in Figure 4.1 and Figure 4.2). The relative positions of these nine image samples in Figure 4.7 match those of their manifold coordinates in Figure 4.5(c); that is, each row of Figure 4.7 corresponds to each nearly-horizontal arrow in Figure 4.5(c), and each column corresponds to each vertical arrow in Figure 4.5(c). From left to right in each row of Figure 4.7, i.e., following each nearly-horizontal arrow in Figure 4.5(c), the variation in the nature of the corresponding image samples is clearly seen to be changes in the spatial decay level of the autocorrelation. Thus, one of the discovered variation patterns represents the variation in the spatial decay level of the autocorrelation (which corresponds to variation in A). Similarly, inspecting image samples from top to bottom in each column of Figure 4.7, i.e., following each vertical arrow in Figure 4.5(c), the variation in the nature of the image samples is clearly seen to be changes in the orientation angle of the autocorrelation. Thus, the other discovered variation pattern represents the variation in the orientation angle of the autocorrelation (which corresponds to variation in γ).

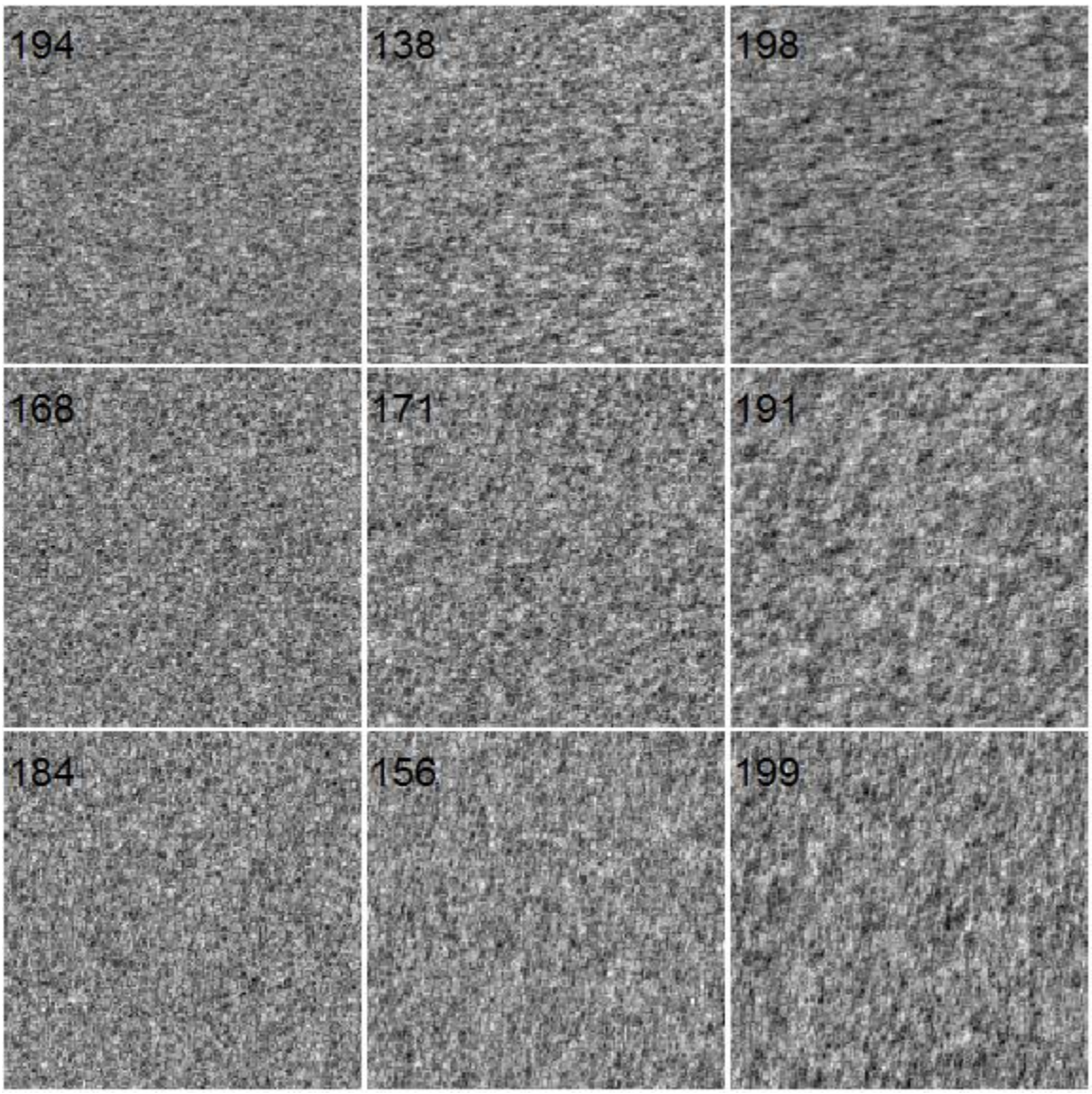


Figure 4.7. Visualization of the two variation patterns identified in the simulation example. The nine image samples have estimated manifold coordinates that lie at the intersections of the six arrows in Figure 4.5(c). The number on each image is its image index, for comparison with Fig 5. From left to right of each row, the systematic variation is variation in the spatial decay level of the autocorrelation. From top to bottom of each column, the systematic variation is variation in the orientation angle of the autocorrelation.

As discussed in Section 4.3, multiple variation patterns existing in the image samples make it difficult to recognize the physical meaning of each variation pattern by simply inspecting the original image samples without any ordering. Our approach provides a means of more clearly

discerning each variation pattern based on inspecting the image samples in an ordering that corresponds to particular paths in the learned manifold coordinate space. This helps identify the nature of the variation and, in a context such as the textile quality control example in the subsequent section, the underlying root-causes of the variation for process improvement. Similarly, in the present simulation example, the image samples could represent surfaces roughness image samples for fabricated metal parts, and variation in the autocorrelation orientation angle and decay rate could represent undesirable sources of variation introduced by faulty tooling or other root causes that could be identified and eliminated.

In our approach, one can use any manifold learning method that takes pairwise dissimilarities as the input. Moreover, we have derived two different dissimilarity measures (KL and AKL) for image samples. In the following, we compare the performance of our method in this simulation example using two different manifold learning methods (MDS and ISOMAP) and using KL and AKL. To serve as a reference point for the comparisons, we also include results using a random ordering approach, which generates (instead of learns) manifold coordinates of image samples using a uniform distribution to generate the coordinates. As a performance measure, we focus on comparing how well the estimated manifold coordinates match the coordinates of the parameters that we used to generate the image samples. For this purpose, we used the Procrustes statistic (Goodall 1991), which lies in the interval [0, 1]. A small Procrustes statistic indicates a good match between the estimated coordinates and the ground-truth, allowing translation, rotation, and/or isometric scaling. We used a Monte Carlo simulation, where on each replicate we generated a different set of N image samples as described earlier in this section. For each set of N image samples, we performed Steps 1 and 2 of our algorithm to obtain the learned manifold coordinates in the same manner as above, except that here we computed both KL and AKL dissimilarities in Step 1 and used both MDS and ISOMAP in Step 2, for comparison.

The first four columns of Table 4.1 show the average (across 10 Monte Carlo replicates) of the

Procrustes statistics computed for the learned manifold coordinates with respect to the parameters $\{A, \gamma\}$ and $\{\phi_1, \phi_2\}$ used to generate the image samples. It can be seen from these columns that in every case, the average Procrustes statistics were quite small (near 0 in some cases), indicating that the estimated manifold coordinates (using either MDS or ISOMAP) are quite accurately matched with the ground-truths of both $\{A, \gamma\}$ and $\{\phi_1, \phi_2\}$. On the other hand, the average Procrustes statistics of the random ordering approach in the last column of Table 4.1 are close to the maximum value of the Procrustes statistic of 1. This means that the coordinates generated by the random ordering approach have no relationship with the ground-truths of either $\{A, \gamma\}$ or $\{\phi_1, \phi_2\}$. Overall, MDS was slightly better than ISOMAP. The performance of the AKL dissimilarity was almost perfect in this example when using MDS. Note that the simulated image samples in this example truly had the same size level of prediction noise, and hence satisfied the assumptions of the AKL dissimilarity measure as discussed in Remark 2.

Table 4.1. Average Procrustes statistics of the estimated manifold coordinates across the Monte Carlo replicates for the simulation example for various versions of our approach (using MDS and ISOMAP with the KL and AKL dissimilarities) and the random ordering approach. The Procrustes statistic compares the estimated manifold coordinates with the ground-truth for both parameterization types $\{A, \gamma\}$ and $\{\phi_1, \phi_2\}$.

Parameterization	KL-MDS	AKL-MDS	KL-ISOMAP	AKL-ISOMAP	Random
$\{A, \gamma\}$	0.120	0.046	0.091	0.054	0.992
$\{\phi_1, \phi_2\}$	0.035	0.005	0.027	0.023	0.992

4.5 Textile Example

In this section, we test our approach on an example involving a set of $N = 100$ image samples with two variation patterns that we created from real textile image samples similarly to the ones in Figure 4.1(a, b). The two variation patterns were obtained by digitally contracting the original image samples by anywhere between 0% and 50% in the horizontal and vertical directions. Note that the horizontal (vertical) contraction makes the fiber strand and gap thickness in the horizontal (vertical) direction decrease while those in the vertical (horizontal) direction stay the same. The

amounts of the horizontal (h) and vertical (v) contraction for each image in this set of 100 image samples were also generated as a stratified random sample using LHS. The generated values of h and v are shown in Figure 4.8(a), in which the numbers represent the image indices. We have released this data set in the **textile3** R data package (Bui and Apley 2019c). Figure 4.9 shows 18 image samples randomly selected from the set of 100 image samples.

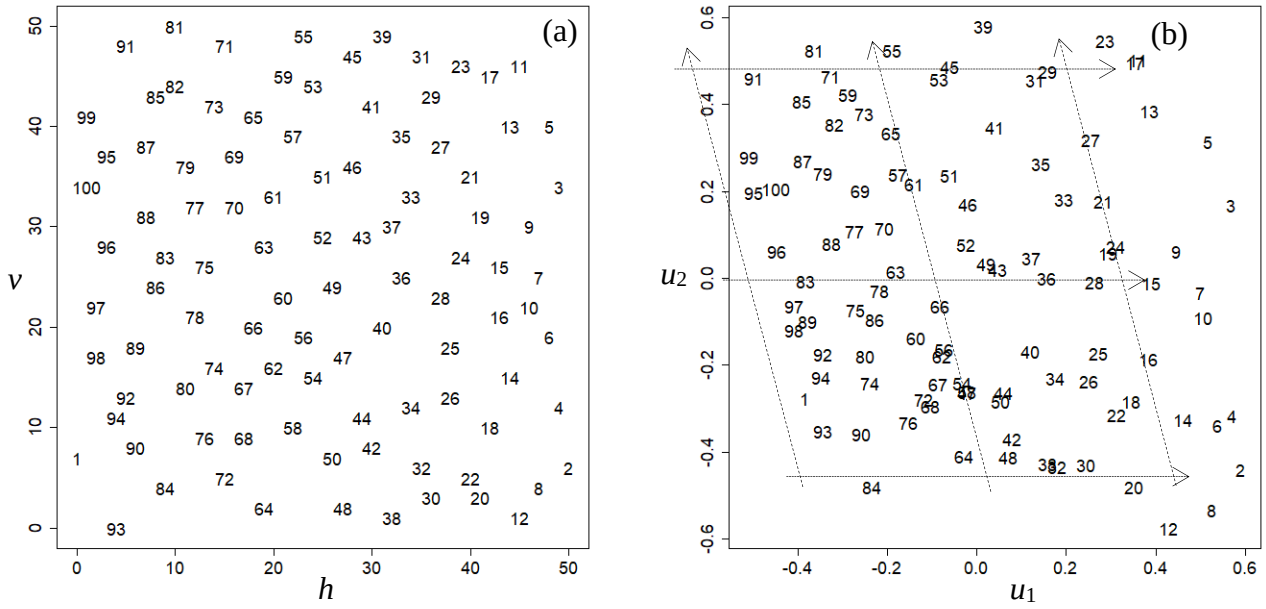


Figure 4.8. Some results for the textile example: (a) Generated amounts of contractions $\{h, v\}$ for the 100 image samples and (b) An ICA version of the estimated coordinates for our approach using MDS and the KL dissimilarity. The numbers in each panel are the image indices. Comparing the image index numbers in (a) and (b) shows that MDS with the KL dissimilarity estimated the true $\{h, v\}$ quite well.

We applied our algorithm to this set of textile image samples, as follows. In Step 1 of our algorithm, to construct the training data when fitting the models for all image samples, we again chose l = 5 using the CV procedure for selecting l in Section 6.2. The same as for the simulation example in Section 4.4, we used regression trees as the fitted supervised learning models $\{\hat{g}_j(\cdot): j = 1, 2, \ldots, N\}$ in this example. The value cp = 0.0001 was also chosen by the CV procedure for selecting cp in Section 6.2 and used to fit the regression tree for each image. After

that, the dissimilarity matrix was computed using the KL dissimilarity.

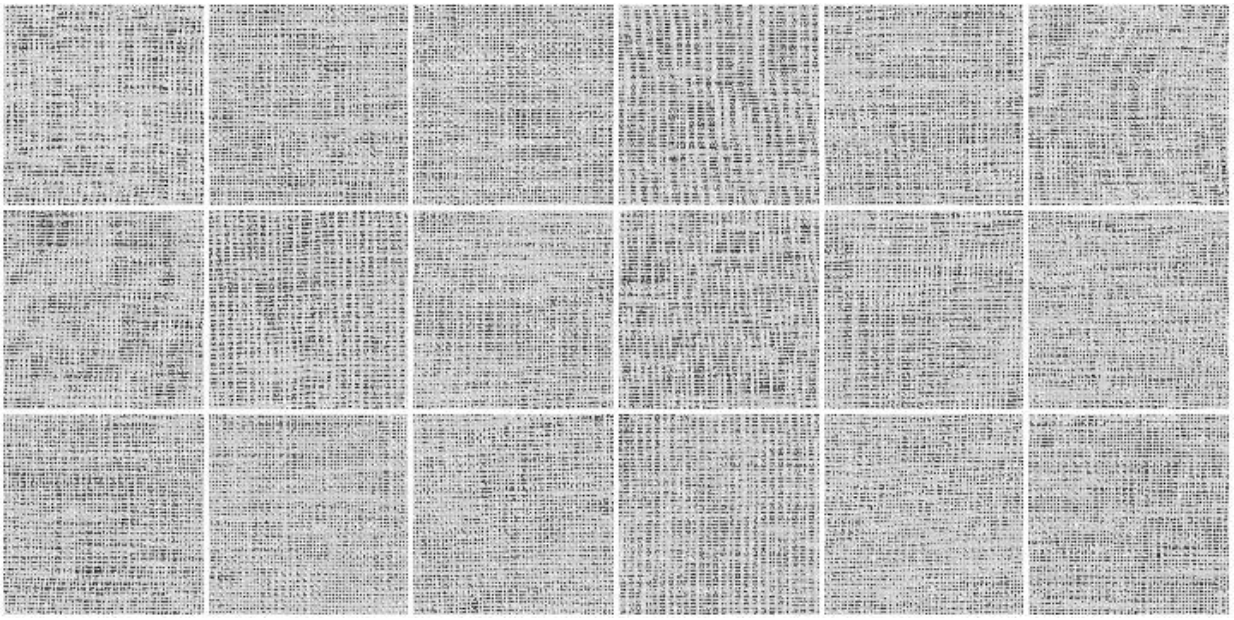


Figure 4.9. A set of 18 textile image samples randomly selected from the set of 100 image samples in the textile example.

In Step 2 of our algorithm, we applied MDS to the dissimilarity matrix obtained in Step 1 to estimate the two-dimensional manifold coordinates for the image samples. The manifold dimension of two was again chosen based on the number of dominant eigenvalues. Figure 4.8(b) plots the estimated two-dimensional coordinates of the image samples after being rotated using independent component analysis (ICA) for visualization purposes. As can be seen in Figure 4.8(b), our approach is able to quite accurately rank the image samples according to the true contraction amounts that were used to generate the image samples.

In Step 3, we visualized the individual effects of the variation patterns present in the image samples based on the estimated manifold coordinates in Figure 4.8(b). Similar to the simulation example in Section 4.4, we inspected the image samples falling closely to each of the six arrows in Figure 4.8(b). Due to limited space, Figure 4.10 shows only nine image samples (again magnified and cropped, to improve visibility) that have the estimated manifold coordinates close

to the values represented by the intersections of the six arrows in Figure 4.8(b). From left to right in each row of Figure 4.10, i.e., following each horizontal arrow in Figure 4.8(b), the thickness of the vertical fiber strands and gaps clearly decreases, while those in the horizontal direction remain approximately the same.

Similarly, from bottom to top in each column of Figure 4.10, i.e., following each vertical arrow in Figure 4.8(b), the thickness of the horizontal fiber strands and gaps in the image samples clearly decreases, while those in the other direction remain roughly unchanged. In this way, the individual physical nature of the two discovered variation patterns can be revealed, and this understanding can aid the users to identify root causes of these variation patterns. We emphasize again that no prior knowledge of the true nature of the variation patterns was incorporated into the algorithm.

As in Section 4.4, we numerically compared (via the Procrustes statistic) the manifold learning results using the KL and AKL dissimilarities and using the random approach. In addition, we compared them with what we call the “FFT-oracle” approach, because it uses advanced knowledge of the nature of the changes to define specific FFT features (described below) that are known to relate to the specific changes in the surfaces. For this reason, this and other feature-based methods are "blind" to general changes that are not changes in the specific features that are monitored. In contrast, our approach does not incorporate any such prior knowledge of the variation patterns, because the goal is to discover new features that govern the (previously unidentified) variation patterns.

The FFT-oracle approach performs the following three steps for each image sample. First, this approach computes the 1D fast Fourier transform (FFT) for each row and column of the image sample and produces a horizontal FFT matrix and a vertical FFT matrix, respectively. In other words, each row in the horizontal FFT matrix is the 1D FFT of the corresponding row in the image sample. Then, the approach computes the average 1D frequency spectrums of the horizontal and vertical FFT matrices over their rows and columns, respectively. Finally, the first-peak frequencies

of the two average 1D FFT spectrums are used as the 2-D coordinates (the first peak frequencies were chosen based on prior knowledge of the nature of the variation patterns, because they correspond to the spacing between textile fibers).

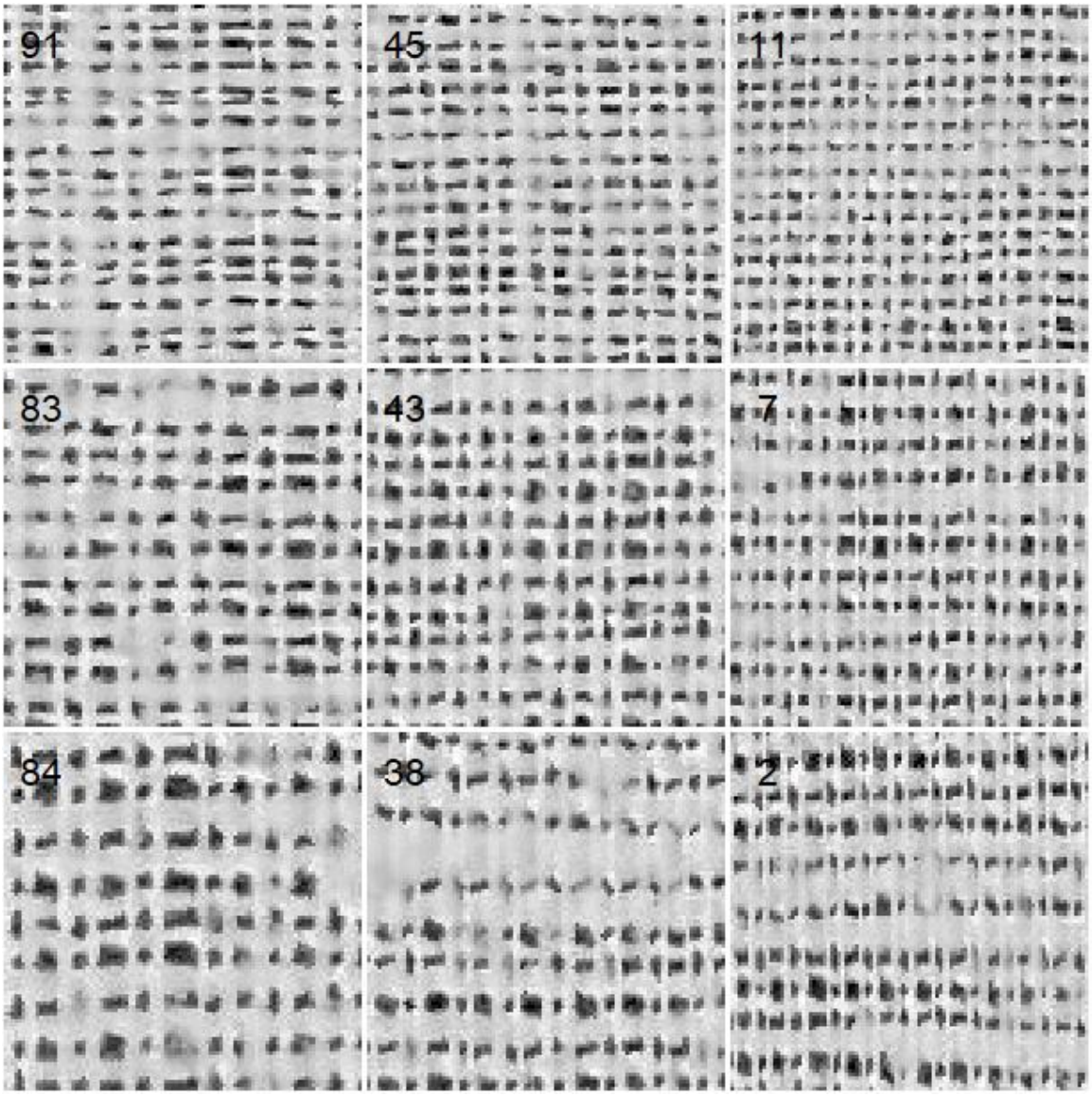


Figure 4.10. Visualization of the two variation patterns identified in the textile example. The nine image samples shown have estimated manifold coordinates that lie at the intersections of the six arrows in Figure 4.8(b). The number on each image is its index. Moving left/right within rows represents systematic variation in the thickness of the vertical fiber strands and gaps, and moving up/down within columns represents systematic variation in the thickness of the horizontal fiber strands and gaps.

As the first-peak frequencies of the two spectrums are almost perfectly correlated with the contraction variation patterns that we created for the textile image samples, this approach can recover the true amounts of contraction almost perfectly. Because our approach is general and does not use specific features based on prior knowledge of the nature of the changes, it is of course not fair to compare FFT-oracle approach with our approach. Rather, we include it because it gives what can be viewed as a performance benchmark for this example.

Table 4.2 shows the comparison using the Procrustes statistic, from which it can be seen that the KL dissimilarity performed better than the AKL dissimilarity when using both manifold learning algorithms (MDS and ISOMAP). The AKL dissimilarity did not work as well as in the simulation example, perhaps because the constant-variance assumption (see Remark 2) did not hold in this example. MDS provided slightly better results than ISOMAP. As in the simulation example in Section 4.4, the Procrustes statistic for the random ordering approach was close to 1 (see Column "Random" in Table 4.2). Interestingly, our approach performed comparably with the hypothetical benchmark FFT-oracle approach, which uses prior knowledge of the nature of the variation.

Table 4.2. Procrustes statistics of the estimated coordinates (in comparison with the ground-truth) for the textile example for various versions of our approach (using MDS and ISOMAP with the KL and AKL dissimilarities), the random ordering approach, and the FFT-oracle approach (which serves only as a hypothetical benchmark, since it uses prior knowledge of the variation patterns).

KL-MDS	AKL-MDS	KL-ISOMAP	AKL-ISOMAP	Random	FFT-oracle
0.038	0.066	0.054	0.075	0.997	0.051

4.6 Further Discussions

4.6.1 Visualization and the role of sorting and decoupling

This section further discusses the visualization step for understanding the nature of individual variation patterns present in the image samples, which is challenging when multiple variation

patterns are concurrently present with their effects mixed together. We also discuss how the manifold learning at its most elemental level can be viewed simply as sorting the image samples according to the level of the pattern(s) and decoupling the effects of multiple patterns, and how this substantially facilitates visualization of the patterns.

Automatic image interpolation methods (and other related techniques such as image blending and morphing) have potential to help visualize the variation patterns by interpolating image samples that fall closely along some path on the learned manifold and potentially could be used to enhance our approach. For example, we have implemented (for brevity, the results are not shown here) the image interpolation method of Darabi et al. (2012) for this purpose. Although it did provide effective visualization of the autocorrelation orientation angle change in the simulation example (by successfully generating fictitious image samples that have autocorrelation orientation angle that changes smoothly), it was unable to provide useful visualization for our textile example. As of yet, we have not been able to get an image interpolation method to work robustly for visualizing the variation. We do think that developing automatic image interpolation, morphing, and blending methods that are applicable to stochastic texture surface (current methods are intended for use with images containing distinct objects that change position, shape, etc.) is a fruitful area for research and would strengthen our approach. However, we leave this for future research for two reasons. The first reason is that it is a challenging problem that will require substantial investigation.

The second reason is that the "sorting" and "decoupling" that are inherent to our current approach, even though it sounds rather mundane on the surface, is actually very helpful in visualizing the nature of the variation (relative to just looking at the image samples in random order). By "sorting" and "decoupling", we mean the following. If we stop our algorithm at Step 2 (i.e., after the manifold learning step), the outcome of our algorithm is the manifold coordinates of the image samples. This can be viewed as sorting (based on distances) the high-dimensional image

samples in a much lower dimensional space (the dimension of which is the number of patterns present in the image samples) according to the level of each pattern. For example, the arrows in Figure 4.8(b), which shows the manifold learning results of the textile example, represent different instances of the sorting process for the textile image samples. As shown in Figure 4.10 and explained in the surrounding discussion in Section 4.5, each horizontal arrow corresponds to a continuous change in the width of the vertical fiber strands and gaps while the width in the horizontal direction is roughly held fixed; each vertical arrow corresponds to a continuous change in the width of the horizontal fiber strands and gaps while the width in the vertical direction is roughly held fixed. Because the sorting of each pattern takes place along a one-dimensional path associated with one manifold coordinate, while the levels of other patterns corresponding to the other manifold coordinates are roughly held fixed, the sorting process also decouples the variation patterns into individual one. For instance, the changes in the thickness of the fiber strands and gaps in both horizontal and vertical directions in the textile image samples in Section 4.5 are decoupled into a change in just the vertical direction or the horizontal direction, after sorting.

The simple act of sorting the image samples according to the level of the pattern (especially in conjunction with decoupling of the patterns, when multiple patterns are present) is very useful in visualization for the following reasons. After sorting, the image samples are ordered according to an individual pattern, from one extreme to the other extreme. If there are multiple patterns present, as described in the previous paragraph the patterns are decoupled in the sense that we are able to roughly fix the levels of the other patterns, so that the individual pattern in question can be isolated with the image samples sorted according to it. This is the basis for Step 3 of our algorithm and what is being done in Figure 4.8 and Figure 4.10, when we visualize a series of image samples that correspond to moving along one of the arrows in Figure 4.8(b). The human brain is much better able to recognize a pattern when (*i*) it is varied continuously, in order, over its full range; and (*ii*) the effects of the individual pattern are isolated/decoupled from the effects of any other patterns

by holding the levels of the other patterns roughly fixed. Sorting can also help to visually identify clusters of similar image samples and/or outlying image samples. If the image variation represents manufacturing variation, then identifying clusters/outliers can help in discovering abnormal behavior in the manufacturing process.

Moreover, the sorting and decoupling inherent to manifold learning is even more important in high dimension (e.g., when there are three or more variation patterns present). This is because the manifold learning finds a parameterization for the set of variation patterns, and then, allows one to individually inspect each variation pattern one-by-one in the parameterization space. The higher the dimension, the more difficult it is for manual inspection (i.e., without using our sorting and decoupling approach).

4.6.2 Choice of Neighborhood Size and Complexity Parameter

This section discusses how the neighborhood size l and complexity parameter cp of the regression trees can be chosen using CV (the complexity parameters of other supervised learning models can be chosen using the same principle). We demonstrate with the textile example in Section 4.5. To investigate how the results would change as l and cp are varied, we also compare the manifold learning results (based on the Procrustes statistic) for different combination of l and cp values.

To select the neighborhood size l, we recommend the following procedure based on CV. First, we randomly select a small set of image samples and fit supervised learning models for each of them using a range of l values. Then, we plot the CV R^2 of the fitted models against l and select the smallest l value at which the CV R^2 stabilizes for all samples. For example, Figure 4.11 shows such a plot for three image samples (each of which corresponds to a curve in Figure 4.11) that were randomly selected from the textile image samples in Section 4.5. It can be seen from Figure 4.11 that $l = 5$ is roughly the smallest value after which the CV R^2 plateaus in each curve, and thus, this value was chosen for our example in Section 4.5. Choosing the smallest l value after which

CV R^2 roughly plateaus results in somewhat less complex models than if one attempted to maximize CV R^2, and less complex models have computational (and perhaps robustness) advantages. Note that in other cases, users may need to inspect the CV R^2 plots for more than the three image samples shown in Figure 4.11.

A similar CV procedure can be used to select the common complexity parameter *cp* for the regression trees. Based on plots of CV R^2 against *cp* of some models fitted to a small set of image samples, we select the smallest *cp* value after which the CV R^2 plateaus (a smaller *cp* will result in a more complex tree). For instance, Figure 4.12 plots the CV R^2against *cp* (*cp* decreases from left to right in the plot) for a model fitted to one of the three image samples selected for the analysis in Figure 4.11, and shows that the CV R^2 plateaus around *cp* = 0.0001; such plots for the other two image samples selected for the analysis in Figure 4.11 demonstrate similar behavior, and hence, are not shown here. Therefore, we selected *cp* = 0.0001 for our textile example in Section 4.5.

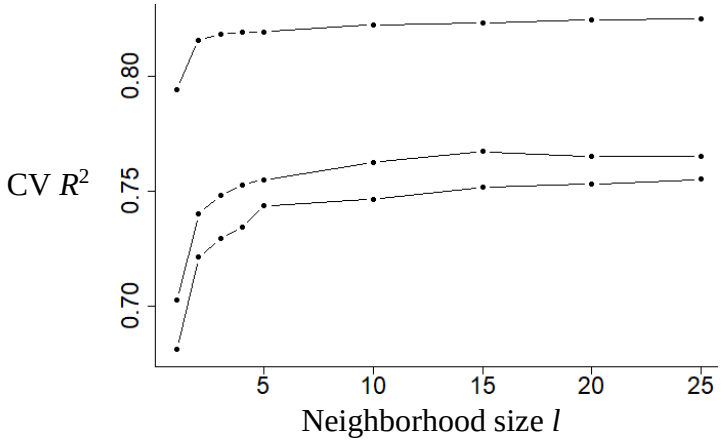


Figure 4.11. CV R^2 versus neighborhood size *l* for the models fitted to three textile image samples (each of which corresponds to a curve in the plot) randomly selected from the image samples in Section 4.5.

Remark 6: We have found that choosing a common *cp* parameter (i.e., the same *cp* value is used for the tree fitted to each image sample) generally works a little better than choosing separate *cp* parameters for each image; at least, this was the case in all of our examples (in terms of the

Procrustes statistic performance measure). We suspect this is because the former is a little more robust than the latter to selecting a poor *cp* due to random chance and image-to-image variation, and while there is variation in the nature of the image samples, the variation is not so large that it warrants using a different *cp* for different image samples. Hence, we used the former approach in this chapter, and also suggest it for choosing the complexity parameters of other supervised learning algorithms (if something other than a regression tree is used).

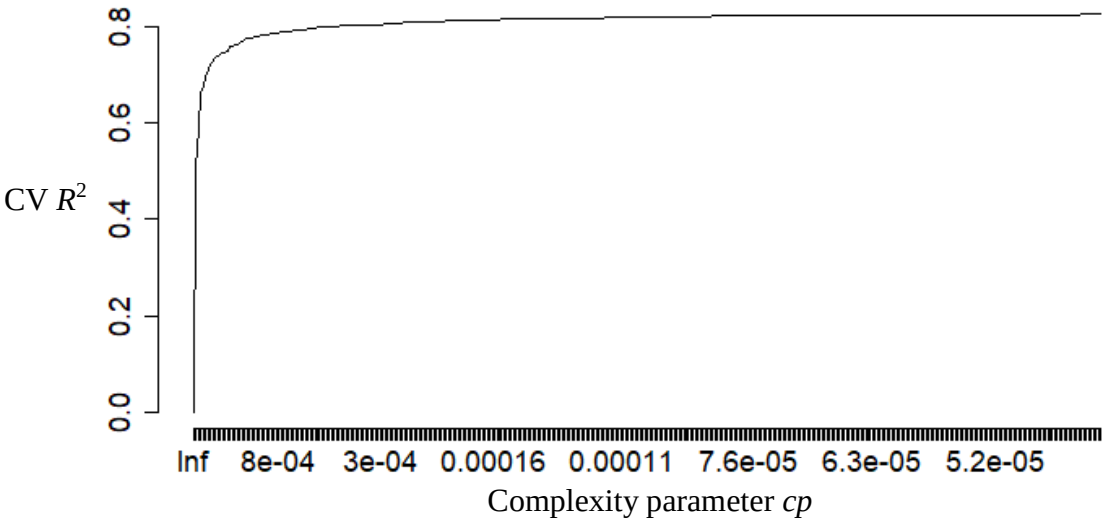


Figure 4.12. CV R^2 versus complexity parameter *cp* for one of the models fitted to three textile image samples randomly selected from the image samples in Section 4.5.

To see how the manifold learning results vary with different choices of *l* and *cp*, we compare the Procrustes statistics for the textile example in Section 4.5 using MDS with the KL and AKL dissimilarity measures for different combinations of *l* and *cp* in Table 4.3. It can be seen from Table 4.3 that our above strategy for choosing *l* and *cp* (chosen values are indicated by cells with bold font in Table 4.3) provided good manifold learning results that are comparable to those of the FFT-oracle approach in Section 4.5.

Table 4.3. Procrustes statistics of the estimated coordinates (in comparison with the ground-truth) for the textile example using MDS with the KL and AKL dissimilarities for different combinations of the neighborhood size *l* and the tree complexity parameter *cp*. The numbers in bold indicate the selected values of *l* and *cp* using our CV procedure.

cp	Dissimilarity Measure	Neighborhood size *l*						
		1	2	3	4	5	10	15
0.0001	KL	0.049	0.041	0.044	0.040	**0.038**	0.044	0.056
	AKL	0.519	0.090	0.086	0.072	**0.066**	0.077	0.110
0.001	KL	0.056	0.055	0.056	0.051	0.049	0.056	0.077
	AKL	0.198	0.118	0.112	0.096	0.086	0.102	0.154

4.6.3 Choice of Dissimilarity Measures and Manifold Learning Algorithm

We have derived the KL and AKL dissimilarity measures and tried different manifold learning algorithms (MDS and ISOMAP) in the previous examples. This section discusses how to choose the best combination of the dissimilarity measure and manifold learning algorithm for the visualization step.

In our previous examples, the KL dissimilarity measure consistently worked well (near the top) across all examples, and it worked better for the real example. Hence, it is our recommended measure. However, the AKL dissimilarity measure is still useful for the following two reasons. First, it did perform better than the KL dissimilarity measure when its constant-variance assumption was met (in the simulation example in Section 4.4). Second, the KL and AKL dissimilarities can be computed at the same time with little extra computational effort. Computing both measures allows users to conduct the visualization step for both measures and then select the measure that provides more interpretable results as explained below.

As with dissimilarity measures, users can also try different manifold learning algorithms, and then select the algorithm that provides the most interpretable results. It may be reasonable to start with MDS, because it worked better in our examples. In general, we recommend that users begin the visualization step with KL and MDS. If the results are not very interpretable or the user wants to run the analysis in a different way to see if the results are different (e.g., if the true manifold is highly nonlinear, then ISOMAP should be more effective than MDS as the manifold learning

method; or if the constant-variance assumption of the AKL measure is satisfied, the AKL measure may yield better results than the KL measure), they can always try other combinations to see if the results are more interpretable. This is similar to the standard practice in factor analysis, in which users try various methods like regular PCA, varimax rotation, equimax rotation, promax rotation, etc., to see if one of the methods produces more interpretable results (for details of these methods, e.g., see Johnson and Wichern 2007).

4.7 Summary and Concluding Remarks

In this chapter, we develop an exploratory analysis approach to identify and understand previously unknown systematic variation patterns in the stochastic nature across a set of stochastic textured surface samples. Such data occur commonly in practice, although the problem has not previously been addressed in the literature. We formulate and derive two new pairwise dissimilarity measures (KL and AKL) between the stochastic textured surface image samples. To these new pairwise dissimilarity measures for stochastic textured surface samples, we apply a form of manifold learning that takes dissimilarities as the input to help discover a low-dimensional parameterization of the surface variation present in the given samples. Varying the manifold parameters and visualizing how the surfaces change accordingly helps build an understanding of the physical nature of each systematic variation pattern.

We illustrate the approach with simulation and textile examples. In both examples, our approach was able to accurately identify the manifold parameterization of the variation without incorporating any prior knowledge of the variation. Visualization based on the discovered manifold coordinates allows the physical nature of each individual variation pattern to be better understood. Although we have derived and investigated the KL and AKL dissimilarity measures as the basis for identifying the nature of variation in stochastic textured surface data, they can also be used in other contexts such as classification, characterization, clustering, and outlier detection that involve stochastic textured surface data.

REFERENCES

Anderson, T.W., and Darling, D.A. (1954) A Test of Goodness of Fit. *Journal of the American Statistical Association*, 49, 765-769.

Apley, D. W. and Lee, H. Y. (2003) Identifying spatial variation patterns in multivariate manufacturing processes. *Technometrics*, 45, 220–234.

Apley, D. W. and Shi, J. (2001) A factor-analysis method for diagnosing variability in multivariate manufacturing processes. *Technometrics*, 43, 84–95.

Apley, D. W. and Zhang, F. (2007) Identifying and visualizing nonlinear variation patterns in multivariate manufacturing data. *IIE Transactions*, 39, 691–701.

Armingol, J.M., Otamendi, J., Escalera, A. de la, Pastor, J.M., and Rodriguez, F.J. (2003) Statistical Pattern Modeling in Vision-Based Quality Control Systems. *Journal of Intelligent and Robotic Systems*, 37, 321-336.

Bharati, M.H., and MacGregor, J.F. (1998) Multivariate Image Analysis for Real-Time Process Monitoring and Control. *Industrial and Engineering Chemistry Research*, 37, 4715-4724.

Bharati, M. H., MacGregor, J. F., and Tropper, W. (2003) Softwood Lumber Grading through On-line Multivariate Image Analysis Techniques. *Industrial & Engineering Chemistry Research*, 42, 5345–5353.

Bostanabad, R., Bui, A.T., Xie, W., Apley, D. W., and Chen, W. (2016) Stochastic microstructure characterization and reconstruction via supervised learning. *Acta Materialia*, 103, 89–102.

Box, G.E.P., and Pierce, D.A. (1970) Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 65, 1509-1526.

Bui, A.T., and Apley, D.W. (2017a) **textile**: Textile Images. R package version 0.1.2. https://cran.r-project.org/package=textile

—— (2017b) Textile Images 2. Mendeley Data, v1. http://dx.doi.org/10.17632/wy3pndgpcx.1.

——(2018a) A monitoring and diagnostic approach for stochastic textured surfaces. *Technometrics*, 60, 1–13.

——(2018b) Monitoring for changes in the nature of stochastic textured surfaces. *Journal of Quality Technology*, 50, 363–378.

——(2019a) "An Exploratory Analysis Approach for Understanding Variation in Stochastic Textured Surfaces", *Computational Statistics & Data Analysis*, 137, 33−50.

——(2019b) **spc4sts**: Statistical Process Control for Stochastic Textured Surfaces in R, submitted.

——(2019c) "Textile Image 3", figshare, http://dx.doi.org/10.6084/m9.figshare.7619351.v1.

Chang, S.I., and Yadama, S. (2010) Statistical Process Control for Monitoring Non-linear Profiles Using Wavelet Filtering and B-Spline Approximation. *International Journal of Production Research*, 48, 1049–1068.

Chicken, E., Pignatiello, J.J., and Simpson, J.R. (2009) Statistical Process Monitoring of Nonlinear Profiles Using Wavelets. *Journal of Quality Technology*, 41, 198–212.

Cover, T.M., and Thomas, J.A. (2006) *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ.

Darabi, S., Shechtman, E., Barnes, C., Goldman, D. B., and Sen, P. (2012) Image melding: combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics*, 31, 82–82.

DeCost, B. L. and Holm, E. A. (2017) Characterizing powder materials using keypoint-based computer vision methods. *Computational Materials Science*, 126, 438–445.

Efros, A. A. and Leung, T. K. (1999) Texture synthesis by non-parametric sampling. *In the Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1033–1038.

Goodall, C. (1991) Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society Series B (Methodological)*, 53, 285–339.

Grasso, M., Menafoglio, A., Colosimo, B.M., and Secchi, P. (2016) Using Curve-Registration Information for Profile Monitoring. *Journal of Quality Technology*, 48, 99–127.

Human, S. W., Chakraborti, S., and Smit, C. F. (2010) Shewhart-type control charts for variation in phase I data analysis. *Computational Statistics & Data Analysis* 54, 863–874.

Izenman, A. J. (2013) *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, Springer, New York, NY.

Jensen, W.A., and Birch, J.B. (2009) Profile Monitoring via Nonlinear Mixed Models. *Journal*

of Quality Technology, 41, 18–34.

Jiang, B.C., and Jiang, S.J. (1998) Machine Vision Based Inspection of Oil Seals. *Journal of Manufacturing Systems*, 17, 159–166.

Jiang, M., Zhang, S., Huang, J., Yang, L., and Metaxas, D.N. (2015) Joint Kernel-Based Supervised Hashing for Scalable Histopathological Image Analysis. *Medical Image Computing and Computer-Assisted Intervention*, 9351, 366–373.

Jin, J., and Shi, J. (1999) Feature-Preserving Data Compression of Stamping Tonnage Information Using Wavelets. *Technometrics*, 41, 327–339.

Johnson, R. A. and Wichern, D. W. (2007) *Applied Multivariate Statistical Analysis*, Pearson, Upper Saddle River, NJ.

Lee, H. Y. and Apley, D. W. (2004) Diagnosing manufacturing variation using second-order and fourth-order statistics. *International Journal of Flexible Manufacturing Systems*, 16, 45–64.

Levina, E. and Bickel, P. J. (2006) Texture synthesis and nonparametric resampling of random fields. *The Annals of Statistics*, 34, 1751–1773.

Lin, H.-D. (2007a) Computer-aided visual inspection of surface defects in ceramic capacitor chips. *Journal of Materials Processing Technology*, 189, 19–25.

——(2007b) Automated visual inspection of ripple defects using wavelet characteristic based multivariate statistical approach. *Image and Vision Computing*, 25, 1785-1801.

Liu, J. J. and MacGregor, J. F. (2006) Estimation and monitoring of product aesthetics: application to manufacturing of "engineered stone" countertops. *Machine Vision and Applications*, 16, 374–383.

Liang, Y.-T., and Chiou, Y.-C. (2008) Vision-based Automatic Tool Wear Monitoring System. *In the 7th World Congress on Intelligent Control and Automation*, 6031–6035.

Liu, X. and Shapiro, V. (2015) Random heterogeneous materials via texture synthesis. *Computational Materials Science*, 99, 177–189.

Lou, X., Fiaschi, L., Koethe, U., and Hamprecht, F. A. (2012) Quality Classification of Microscopic Imagery with Weakly Supervised Learning. *Machine Learning in Medical Imaging*, 7588, 176–183.

Lyu, J., and Chen, M. (2009) Automated Visual Inspection Expert System for Multivariate Statistical Process Control Chart. *Expert Systems with Applications*, 36, 5113–5118.

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, 83–98.

Megahed, F.M., and Camelio, J.A. (2012), "Real-time Fault Detection in Manufacturing Environments Using Face Recognition Techniques," *Journal of Intelligent Manufacturing* 23, 393-408.

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, 967–980.

Montgomery, D.C. (2009) *Introduction to Statistical Quality Control*, Wiley, NJ.

Paynabar, K., Jin, J., and Pacella, M. (2013) Monitoring and Diagnosis of Multichannel Nonlinear Profile Variations using Uncorrelated Multilinear Principal Component Analysis. *IIE Transactions*, 45, 1235–1247.

Paynabar, K., Zou, C., and Qiu, P. (2016) A Change-Point Approach for Phase-I Analysis in Multivariate Profile Monitoring and Diagnosis. *Technometrics*, 58, 191–204.

Qiu, P. (1998) Discontinuous Regression Surfaces Fitting. *The Annals of Statistics*, 26, 2218–2245.

Qiu, P., and Xing, C. (2013a) On Nonparametric Image Registration, *Technometrics*, 55, 174-188.

——(2013b) Feature based image registration using non-degenerate pixels. *Signal Processing*, 93, 706-720.

Qiu, P. and Yandell, B. (1997) Jump Detection in Regression Surfaces. *Journal of Computational and Graphical Statistics*, 6, 332–54.

Qiu, P. and Zou, C. (2010) Control Chart for Monitoring Nonparametric Profiles with Arbitrary Design. *Statistica Sinica*, *20*, 1655–1682.

Qiu, P., Zou, C., and Wang, Z. (2010) Nonparametric Profile Monitoring by Mixed Effects Modeling. *Technometrics*, 52, 265-277.

Rasmussen, C.E., and Williams, C.K.I. (2006) *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA.

Shan, X. and Apley, D. W. (2008) Blind identification of manufacturing variation patterns by combining source separation criteria. *Technometrics*, 50, 332–343.

Shi, Z., Apley, D. W., and Runger, G.C. (2016) Discovering the nature of variation in nonlinear profile data. *Technometrics*, 58, 371–382.

Song, J. and Kang, J. (2018) Parameter change tests for ARMA–GARCH models. *Computational Statistics & Data Analysis*, 121, 41–56.

Song, Y., Li, Q., Huang, H., Feng, D., Chen, M., and Cai, W. (2017) Low Dimensional Representation of Fisher Vectors for Microscopy Image Classification. *IEEE Transactions on Medical Imaging*, 36, 1636–1649.

Tan, J., Chang, Z., and Hsieh, F. (1996) Implementation of an automated real-time statistical process controller. *Journal of Food Process Engineering*, 19, 49-61.

Taylor, S.L., and Nunes, M.A. (2014) LS2Wstat: A Multiscale Test of Spatial Stationarity for LS2W processes. R Package Version 2.0.3. https://CRAN.R-project.org/package=LS2Wstat.

Taylor, S.L., Eckley, I.A., and Nunes, M.A. (2014) A Test of Stationarity for Textured Images. Technometrics, 56, 291–301.

Tenenbaum, J. B., Silva, V. de, and Langford, J. C. (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290, 2319–2323.

Therneau T. and Atkinson, B. (2018) **rpart**: Recursive Partitioning and Regression Trees. R package version 4.1.13. https://CRAN.R-project.org/package=rpart.

Torgerson, W. S. (1952) Multidimensional scaling: I. Theory and method. *Psychometrika*, 17, 401–419.

Torquato, S. (2002) Statistical Description of Microstructures. *Annual Review of Materials Research*, 32, 77–111.

Tunák, M., and Linka, A. (2008) Directional Defects in Fabrics. *Research Journal of Textile and Apparel*, 12, 13–22.

Tunák, M., Linka, A., and Volf, P. (2009) Automatic assessing and monitoring of weaving density, *Fibers and Polymers*, 10, 830–6.

Viveros-Aguilera, R., Steiner, S.H., and MacKay, R.J. (2014) Monitoring Product Size and Edging from Bivariate Profile Data. *Journal of Quality Technology*, 46, 199–215.

Wells, L.J., Megahed, F.M., Camelio, J.A., and Woodall, W.H. (2012) A Framework for Variation Visualization and Understanding in Complex Manufacturing Systems. *Journal of Intelligent Manufacturing*, 23, 2025–2036.

Woodall, W.H., Spitzner, D.J., Montgomery, D.C., and Gupta, S. (2004) Using control charts to monitor process and product quality profiles. *Journal of Quality Technology*, 36, 309–320.

Xing, C. and Qiu P. (2011) Intensity-Based Image Registration by Nonparametric Local Smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 2081-2092.

Xu, L., Wang, S., Peng, Y., Morgan, J.P., Reynolds, M.R., and Woodall, W.H. (2012) The Monitoring of Linear Profiles with a GLR Control Chart. *Journal of Quality Technology*, *44*, 348–362.

Xu, Y., Zhu, J.-Y., Chang, E. I.-C., Lai, M., and Tu, Z. (2014) Weakly supervised histopathology cancer image segmentation and classification. *Medical Image Analysis*, 18, 591–604.

Yu, G., Zou, C., and Wang, Z. (2012) Outlier Detection in Functional Observations with Applications to Profile Monitoring. *Technometrics*, 54, 308–318.

Zou, C., Tsung, F., and Wang, Z. (2007) Monitoring General Linear Profiles Using Multivariate Exponentially Weighted Moving Average Schemes. *Technometrics*, 49, 395–408.

Zou, C., Wang, Z., and Tsung, F. (2008) Monitoring Profiles Based on Nonparametric Regression Methods. *Technometrics*, 50, 512–526.

Zou, C., Ning, X., and Tsung, F. (2012) LASSO-based Multivariate Linear Profile Monitoring. *Annals of Operations Research*, 192, 3–19.