

**MULTIVARIATE IMAGE ANALYSIS FOR REAL-TIME PROCESS
MONITORING**

By

MANISH H. BHARATI, B. Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Engineering

McMaster University

© Copyright by Manish H. Bharati, September 1997

MASTER OF ENGINEERING (1997)
(Chemical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Multivariate Image Analysis For Real-Time Process Monitoring

AUTHOR: Manish H. Bharati, B. Eng. (Ryerson Polytechnic University)

SUPERVISOR: Professor J. F. MacGregor

NUMBER OF PAGES: xii, 125

ABSTRACT

In today's technically advanced society the collection and study of digital images has become an important aspect of various off-line applications that range from medical diagnosis to exploring the Martian surface for traces of water. Various industries have recently started moving towards vision based systems to monitor several of their manufacturing processes. Except for some simple on-line applications, these systems are primarily used to analyze the digital images off-line. This thesis is concerned with developing a more powerful on-line digital image analysis technique which links the fields of traditional digital image processing with a recently devised statistically based image analysis method called multivariate image analysis (MIA).

The first part of the thesis introduces the area of traditional digital image processing techniques through a brief literature review of three of its five main classes (image enhancement, restoration, analysis, compression, & synthesis) which contain most of the commonly used operations in this area. This introduction is intended as a starting point for readers who have little background in this field, and as a means of providing sufficient details on these techniques so that they can be used in conjunction with other advanced MIA on-line monitoring operations.

MIA of multispectral digital images using latent variable statistical methods (Multi-Way PCA / PLS) is the main topic covered by the second part of this thesis. After reviewing the basic theory of feature extraction using MIA for off-line analyses, a new technique is introduced that extends these ideas for image analyses in on-line applications. Instead of directly using the updated images themselves to monitor a time-varying process, this new technique uses the latent variable space of the image to monitor the increase or decline in the number of pixels belonging to various features of interest.

The ability to switch between the images and their latent variable space then allows the user to determine the exact spatial locations of any features of interest.

This new method is shown to be ideal for monitoring interesting features from time-varying processes equipped with multispectral sensors. It forms a basis for future on-line industrial process monitoring schemes in those industries that are moving towards automatic vision systems using multispectral digital imagery.

ACKNOWLEDGMENTS

I would like to express my greatest gratitude to my loving parents, Harish C. Bharati and Devika H. Bharati, for the support they have provided me throughout my life.

I would also like to thank my supervisor, Dr. John F. MacGregor, for his encouragement, and continuous guidance throughout the course of this work. I also extend my thanks to Dr. J. Fraser Forbes, for the advice and support he provided which enabled me to undertake graduate studies at McMaster University.

My sincere acknowledgments are extended to Professor Paul Geladi, for providing the multivariate image data, without which, this work could not have been possible.

Much thanks also go to my colleagues, with whom I have spent a large portion of the past two years of my life. Special thanks go to Ivan Miletic, Tracy-Clarke Pringle, François Boudreau, Athanassios Kassidas, Bhupinder Dayal, Steve Roney, Ali Esmaili, Jean-François Dube, and Carl Duchesne for their insight and lively discussions on almost any topic.

Finally, I would like to express my deepest gratitude to my aunt, Dr. Renuka G. Swami. I am forever indebted to her for the continuous love and strength she has given me in all stages of my life. It is to her that I dedicate this work.

TABLE OF CONTENTS

| | |
|--|------|
| Abstract | iii |
| Acknowledgments | v |
| List of Figures | viii |
| List of Tables | xii |
| | |
| 1.0 Introduction | 1 |
| 2.0 Traditional Image Processing Techniques | 6 |
| 2.1 The Digital Image | 8 |
| 2.1.1 Intensity (Grayscale) Image | 9 |
| 2.1.2 Binary Image | 10 |
| 2.1.3 Indexed Image | 10 |
| 2.1.4 RGB Image | 11 |
| 2.2 Image Enhancement and Restoration | 11 |
| 2.2.1 Image Histogram and Histogram Equalization | 13 |
| 2.2.2 Image Adjustment | 15 |
| 2.2.3 Filtering | 17 |
| 2.2.3.1 Filtering in Spatial Domain | 18 |
| 2.2.3.2 Filtering in Frequency Domain | 31 |
| 2.3 Image Analysis | 39 |
| 2.3.1 Segmentation | 40 |
| 2.3.2 Morphological Operations | 44 |
| 2.3.3 Object Classification and Measurement | 59 |

| | |
|--|------------|
| 3.0 Multivariate Image Analysis (MIA) Using Statistical Latent Variable | |
| Methods | 65 |
| 3.1 Overview of Multivariate Image Analysis | 67 |
| 3.2 Real-Time Monitoring Using MIA | 89 |
| 3.2.1 Extension of MIA Technique for Real-Time Monitoring | 91 |
| 3.2.2 Example of Feature Monitoring using MIA: Remote Sensing | 96 |
| 3.3 Conclusions | 110 |
| 4.0 Summary And Conclusions | 112 |
| 4.1 Directions for Future Research | 118 |
| 4.1.1 Pattern Recognition & Image Classification using MIA | 119 |
| 4.1.2 Combining Traditional Image Processing Techniques with MIA | 119 |
| 4.1.3 MIA using Regression Model Based Latent Variable Methods | 120 |
| 4.1.4 Industrial Application of MIA On-Line Monitoring Scheme | 120 |
| 4.1.5 Feedback Control Schemes using MIA | 121 |
| References | 122 |

LIST OF FIGURES

| | | |
|-------------|---|----|
| Figure 2.1 | A 10 x 10 subsection (shown as a white square) of an 8-bit intensity image represented as a 10 x 10 array of intensity values | 9 |
| Figure 2.2 | Use of image histogram and histogram equalization techniques to enhance the contrast of an 8-bit grayscale image | 15 |
| Figure 2.3 | Use of image adjustment (input cropping) to enhance the contrast of a degraded intensity (grayscale) image | 16 |
| Figure 2.4 | Use of a 3 x 3 moving average filter for smoothing and unsharp masking enhancement | 21 |
| Figure 2.5 | Gaussian low-pass filtering ($\sigma^2 = 0.49$) with 3 x 3 convolution mask to smooth an 8-bit intensity image | 22 |
| Figure 2.6 | Use of a moving median filter to remove outlier pixels from an 8-bit intensity image | 24 |
| Figure 2.7 | Use of a moving median filter to remove sensor induced periodic noise from an 8-bit intensity image | 25 |
| Figure 2.8 | Use of a Wiener filter with a convolution mask of dimension 3 x 3 to both smooth noise and preserve edges in an 8-bit intensity image | 27 |
| Figure 2.9 | Use of a 3 x 3 high-pass spatial filter to sharpen the appearance of an 8-bit intensity image | 28 |
| Figure 2.10 | Use of first-derivative vertical and horizontal edge detection filters to enhance an 8-bit intensity image | 29 |
| Figure 2.11 | Use of a 3 x 3 Laplacian filter ($\alpha = 0$) to detect edges in all directions of an 8-bit intensity image | 31 |

| | | |
|-------------|---|----|
| Figure 2.12 | Use of 2D-FFT to display spatial frequencies and their magnitudes of an 8-bit intensity image | 33 |
| Figure 2.13 | Use of 2D-FFT to identify the frequency component of sensor induced periodic noise patterns in an 8-bit intensity image | 35 |
| Figure 2.14 | Use of a frequency domain band-reject filter to attenuate sensor induced noise frequency patterns from an 8-bit intensity image | 38 |
| Figure 2.15 | Thresholding used to create 2 categories at $t = 0.50$ on an 8-bit intensity image to segment bacterial growth on photographic film from the background | 41 |
| Figure 2.16 | Use of the 'intermeans algorithm' to calculate threshold value $t = 0.4734$ for segmenting an 8-bit intensity image | 43 |
| Figure 2.17 | Examples of morphological erosion with various structuring elements; (a) Horizontal line, (b) Square, (c) Disc | 48 |
| Figure 2.18 | Examples of morphological dilation with various structuring elements; (a) Horizontal line, (b) Square, (c) Disc | 50 |
| Figure 2.19 | Examples of opening operations using various shapes of structuring elements; (a) Horizontal line, (b) Square, (c) Disc | 52 |
| Figure 2.20 | Examples of closing operations using various shapes of structuring elements; (a) Horizontal line, (b) Square, (c) Disc | 53 |
| Figure 2.21 | Example of grayscale erosion using a square structuring element | 55 |
| Figure 2.22 | Example of grayscale dilation using a square structuring element | 57 |
| Figure 2.23 | Example of grayscale opening using a square structuring element | 58 |
| Figure 2.24 | Example of grayscale closing using a square structuring element | 59 |
| Figure 3.1 | A stack of 4 congruent images (collected in 4 different wavelengths) to form a 512 x 512 x 4 Multivariate Image | 67 |

| | | |
|-------------|---|----|
| Figure 3.2 | A 512 x 512 x 4 Multivariate Image viewed as a two-way array of 4 x 1 variable vectors in the image plane. For graphical clarity not all variable vectors are shown. | 68 |
| Figure 3.3 | A 512 x 512 x 4 multivariate image reorganized into a (512 · 512) x 4 array followed by PCA decomposition into reduced dimensional sub-space | 71 |
| Figure 3.4 | A 512 x 512 x 4 LANDSAT (MSS) satellite image. The 4 wavelength bands (variables) are: (a) 500-600 nm {green-yellow}, (b) 600-700 nm {orange-red}, (c) 700-800 nm {deep red}, (d) 800-1100 nm {near-infrared} | 76 |
| Figure 3.5 | The four score images of the 512 x 512 x 4 satellite multivariate image: (a) T_1 , (b) T_2 , (c) T_3 , (d) T_4 | 78 |
| Figure 3.6 | Color coded score plots of the satellite multivariate image: (a) $t_1 - t_2$, (b) $t_2 - t_3$, (c) $t_1 - t_3$, (d) $t_1 - t_4$ | 82 |
| Figure 3.7 | Detailed $t_1 - t_2$ score plot of the satellite multivariate image with a class- masking of the upper-half of lower dense pixel cluster (masked in maroon) | 85 |
| Figure 3.8 | Overlay of T_1 image with highlighted pixels from class outlined in figure 3.7 | 86 |
| Figure 3.9 | Latent variable decomposition of an unfolded training multivariate image to develop a PCA training model | 93 |
| Figure 3.10 | Training image (outlined by white box) chosen from full scene for calibration of PCA training model | 98 |
| Figure 3.11 | Detailed t_1-t_2 score plot of the training multivariate image with color coded pixel intensities | 99 |

| | | |
|-------------|--|-----|
| Figure 3.12 | Detailed t_1 - t_2 score plot of training image with highlighted masks that adequately represent the four features of interest | 100 |
| Figure 3.13 | Three of nine (scan 1, 5, 9) input images outlined as white boxes on false color T_1 full scene score image with corresponding color coded t_1 - t_2 score plot images | 102 |
| Figure 3.14 | Bar chart of total number of pixels belonging to deep water areas through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9 | 104 |
| Figure 3.15 | Bar chart of total number of pixels belonging to shallow water areas through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9 | 105 |
| Figure 3.16 | Bar chart of total number of pixels belonging to major roads & paved areas through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9 | 106 |
| Figure 3.17 | Bar chart of total number of pixels belonging to major golf courses through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9 | 107 |

LIST OF TABLES

| | | |
|------------------|--|------------|
| Table 3.1 | Mapping of Intensity Ranges from 3D Histogram Intensity Matrix to Colors | 81 |
| Table 3.2 | Average Times (over 9 scans) to Monitor 4 Features of Interest from Satellite Multivariate Image Sequence | 109 |

1. INTRODUCTION

The field of digital image processing has existed for the past four decades; however, it has been the recent advancements in video and computer technology over the past 10 years that have made this field blossom. Various image processing operations that previously took days can now be performed within fractions of seconds using today's high speed computers. Due to the present technological revolution, computational speeds as well as their affordability are steadily increasing. Various advanced digital imaging sensors are also experiencing similar trends. As a result, computationally driven digital image processing techniques equipped with advanced digital imaging sensors are being introduced in various fields. These techniques are currently applied in such fields as remote sensing, agriculture, microscopy, medicine, biology, chemistry, space exploration etc. Furthermore, an increasing number of new applications for these techniques in various other fields are being conceived on a regular basis. Thus, it is becoming clear that the future of several technological fields lies in automated visually based systems that are governed by digital image processing techniques.

Over the past decade the field of digital image processing has made a significant impact on various industries. Several industrial processes are rapidly moving towards automatic vision systems to monitor their processes both off-and-on-line with digital image processing techniques playing a significant role. Previously existent sensory equipment is either being replaced or enhanced with technically advanced digital imaging sensors to help monitor these processes. Examples include the use of digital cameras to monitor the state of combustion in furnaces, laser induced scanners to monitor surface properties and other faults in sheet-forming processes. Customized digital image processing techniques are used to analyze thousands of images being intermittently

transmitted by these sensors to detect faults and monitor several time-varying industrial processes. However, these techniques are usually restricted to simple image processing operations in various time-varying industrial processes, mainly due to the need for quick analysis results after acquiring each image from the sequence.

Recently, the digital image sensing technology has introduced various technically advanced digital sensors that have the ability to acquire digital images at multiple wavelengths. These sensors have the advantage of acquiring more information regarding the scene being depicted as compared to simple digital imaging sensors. This is because light in different wavelengths of the spectra enhances unique features that are wavelength-specific. As a result, a multispectral digital imaging sensor collects a set of wavelength-specific digital images which, when pooled together, form a multispectral digital image. However, due to the presence of multiple digital images, a multispectral imaging sensor transmits a lot more data to the image analysis system than do conventional digital imaging sensors.

The main objective of this thesis is to create an efficient link between the fields of digital image processing and industrial process monitoring with the help of latent variable statistical methods (Multi-way PCA / PLS). This will be attempted by testing the use of a recently developed image analysis technique called Multivariate Image Analysis (MIA). This statistically based technique is used for monitoring time-varying processes using digital multispectral sensors. This research project is aimed towards developing some of the basic methods that can eventually be used for devising statistical process control schemes and feedback control schemes of various industrial processes that are equipped with technically advanced automatic vision systems. This thesis is intended as a first step towards this goal. To eventually devise an image based industrial process control scheme through MIA, one must have a firm understanding of traditional digital image processing. Only after the most commonly used operations in this field are soundly understood can the potential of using these techniques be realized. Therefore, a brief literature review of traditional digital image processing is included. This will provide

future researchers in this project, who have very little background in this area, with a concise introduction, as well as ample starting points for their research goals.

To analyze multispectral digital images, an overview of the theory behind digital image analysis using latent variable statistical methods (MIA) is presented. In the MIA literature a multispectral digital image is categorized as a multivariate image where the different wavelengths are the variables. The majority of the basic theory and applications of MIA have concentrated on off-line analysis of multivariate images to extract features of interest, and highlight their corresponding pixels. This is accomplished with the help of score and loading vectors which form the reduced dimensional latent variable space of a multivariate image. To apply the MIA techniques for process monitoring and feedback control, the potential of extending its ideas to on-line applications needs to be explored. This may be evaluated by successfully tracking features of interest from a sequence of multivariate images that depict a time-varying process. The development of a technique that extends the main ideas of MIA for on-line feature monitoring through sequences of multivariate images, forms the main contribution of this thesis.

Since this thesis is the first step towards attaining the main goals of the research project, there existed a need for a sound software infrastructure to be developed. The chosen software needed to be both advanced and flexible enough to perform most of the common digital image processing techniques, and at the same time allow the user to develop customized algorithms to perform advanced MIA operations. Various software packages from several digital image analysis companies were evaluated to find a good mix of the above selection criteria. Finally, the Image Processing Toolbox (version 1.0b) for MATLAB® (version 4.2c1) from The Math Works Inc. was chosen as the primary software tool. This toolbox possesses various built-in functions that perform most of the common traditional digital image processing operations on a vast range of digital image types. Furthermore, the generic MATLAB® (version 4.2c1) environment allows various customized matrix operations through manually coded M-Files (MATLAB high-level computer language). Since digital images are nothing more than number arrays

(matrices), this software was ideal for handling the advanced MIA operations which were manually coded through M-Files. Thus, all results obtained throughout this thesis have been produced through MATLAB® (version 4.2c1) equipped with the appropriate toolbox.

This thesis is organized into two main parts: (i) traditional image processing techniques; and (ii) latent variable statistical methods. The main operations of traditional digital image processing are reviewed, and a brief literature review of this field is presented in chapter 2. This chapter introduces the five main classes of digital image processing (image enhancement, restoration, analysis, compression, and synthesis) from which the first three are discussed in further detail. This is complemented by using illustrated examples of the most commonly used operations from these classes on various digital images. These techniques are reviewed to serve a dual purpose. First, to provide an introduction to traditional image processing operations for readers who have little background in this field. This chapter can also provide an insight into various operations that could potentially be used as a part of other image analyses that are concerned with the final goals of this research project. In chapter 3 this thesis employs the use of latent variable statistical methods to analyze multivariate digital images through MIA. This is a recently introduced statistically based digital image analysis technique that has been used to extract features from off-line multivariate digital images.

Chapter 3 first lays down the main concepts of MIA through a brief overview of the theory and applications of this technique. These concepts are illustrated through a feature extraction example of MIA in the remote sensing field by extracting landscape features from a LANDSAT (MSS) multispectral satellite digital image. After laying down the theory and main ideas of MIA, this chapter introduces a new technique that extends these ideas to monitor features of interest from a sequence of multivariate images. This forms the basis of an on-line monitoring scheme that can be implemented in real-time on various time-varying industrial processes equipped with multispectral digital sensors. This scheme is then applied to monitor various landmark features from a

modified version of the multispectral satellite image example. This image was modified to emulate a moving satellite transmitting multispectral images at a particular sampling rate while it moved over a geographical region.

Finally, in chapter 4, several conclusions are drawn regarding the importance of various operations from the two main sections of this report. This chapter also suggests various future directions that may be pursued by researchers in the field of latent variable statistical methods applied to multispectral digital images.

2. TRADITIONAL IMAGE PROCESSING TECHNIQUES

In general terms, *Image Processing* refers to the analysis and manipulation of pictorial information which, in this case, means a two-dimensional visual image. Operations that act to improve, correct, analyze, or in some way alter an image are considered image processing [Baxes, 1994]. There are several types of image processing mechanisms which can be classified into the general fields of optical, analog, and digital image processing. As far as this report is concerned, we will only be focusing on digital image processing.

Due to the recent developments in the digital, computation, and visual sensory fields, the acquisition of visual data has become relatively simple. A growing number of industries and research applications employ the use of digital visual information to automatically perform several tasks. As a result, the field of digital image processing has now gained considerable momentum and is being used in diverse applications. Some of the major users of digital image processing technology include medical/biological research, industries with automatic vision systems, remote sensing, space exploration/astronomy etc. [Baxes, 1994].

Since digital image processing finds use in several different fields, the particular image operation is generally process dependent. One or several combinations of image processing operations can be used to suit the needs of a specific process. While one operation may improve the quality of an image, another may automatically extract information from it. Generally, digital image processing operations can be grouped into five fundamental classes: image enhancement, restoration, analysis, compression, and synthesis [Baxes, 1994]. Depending upon the process, a certain image processing application may use one or several of these classes.

Image enhancement and restoration operations are generally used to improve the quality, or reduce the presence of noise in an image. These operations may be used to either enhance the end result of an image, or as a preprocess to improve the image for subsequent image analysis operations. Primarily, machine vision applications use image analysis as a standard tool for extracting (numerical or graphical) information from visual data. Image analysis operations commonly encompass the segmentation of the image to extract and describe objects within it. This is assisted by the use of several measurement and classification techniques which are an integral part of image analysis operations. Image compression and decompression techniques are used to reduce the data content required to fully describe an image. Images contain a large amount of redundant information which can be eliminated for the purposes of compression, storage, or transportation. The compressed image can later be decompressed with an inverse operation to restore it to its original form. Image synthesis operations are a part of the computational design domain. These operations create images from other image or non-image data. They are also useful when a desired image is either physically impossible to obtain (e.g. MRI, PET images) or does not exist in a physical form (e.g. CAD of airflow in a duct).

This chapter explains in further detail some of the operations that are a part of the first three fundamental classes of image processing. Image compression and synthesis are not described since they have little relevance to using image processing for feature extraction. A large portion of the image processing literature is concerned with the compression and synthesis of digital images. Further details on these operations may be obtained from several sources, some of which are [Baxes, 1994; Ekstrom, 1984; Jähne, 1993].

2.1 The Digital Image

A digital image can be considered as a continuous image which is discretized both in spatial coordinates as well as brightness. A single 2-D digital image usually consists of a rectangular array, which can be thought of as a matrix of color or intensity data. The row and column indices in this matrix identify a pixel (short for *picture element*) which carries a digital value of the light intensity (*brightness*) at that point in the image. Each pixel has an (x, y) coordinate that corresponds to its location within the image. Generally the x coordinate is the pixel's horizontal location (column number), whereas the y coordinate represents its vertical (row number) location. The number of pixels and the range of brightness levels into which a digital image is divided is known as *image resolution*. The spatial resolution of a digital image is determined by the number of pixels used to express an image. Generally, the spatial resolution requirements of an image are based on its intended application. Common dimensions of a digital image matrix range from 320 x 460 to 1024 x 768 pixels. The numeric range of a pixel's brightness is known as the brightness resolution of a digital image. This range is directly dependent upon the number of bits used to digitize the original image. If three bits are used, the brightness can be divided into one of $2^3 = 8$ intensity levels. Depending upon the type and application of the digital image, the number of bits used for digitization range from 1 to 24.

Digital images can be classified into four main classes which are described below. Some image processing techniques are specific to certain types of digital images, thus it is important to understand each class of the digital image. Most commercial image processing software can help in switching between different types of digital images. As far as this chapter is concerned, most of the image processing operations are illustrated using standard grayscale and binary images unless otherwise stated. These images have been acquired from several sources including [Baxes, 1994; Thompson et. al., 1993].

2.1.1 Intensity (Grayscale) Image

This type of digital image is stored as a single array of pixels with each pixel carrying a digital value representing the light intensity (brightness) at the corresponding spatial location in the original image. The intensity values in the array range between 0.0 and 1.0, where 0.0 represents no brightness (i.e. black) and 1.0 represents full brightness (i.e. white). The number of gray levels between black and white are determined by the resolution of the intensity image. Most common grayscale images have a resolution of 8-bits thus dividing the intensity range of the image from black to white into $2^8 = 256$ shades of gray. Figure 2.1 illustrates an 8-bit grayscale image of a building. A 10 x 10 sub-section from this image (represented by the white square) is extracted from the image and is shown as a corresponding 10 x 10 array of intensity values. These values represent the brightness levels that exist in the original grayscale image at their specific locations.

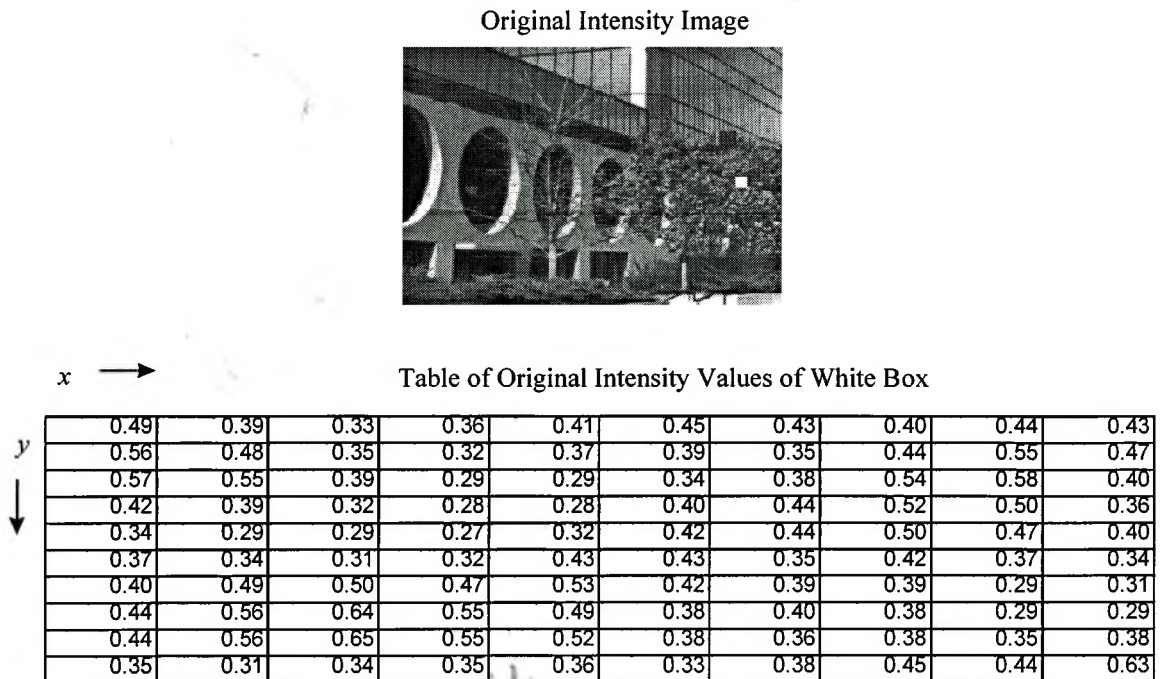


Figure 2.1 A 10 x 10 subsection (shown as a white square) of an 8-bit intensity image represented as a 10 x 10 array of intensity values

2.1.2 Binary Image

A digital binary image is a special category of an intensity image. These images contain only two levels of intensity values which are most commonly represented by 0.0 (black) and 1.0 (white). As a result, the resolution of a binary image is simply 1-bit. Binary images are useful for several image analysis operations which will be discussed later in this chapter.

2.1.3 Indexed Image

This type of digital image data is commonly used to represent color images. Typically color images require more than one digital array when they are digitized for storage by computers. An indexed digital image requires two arrays: a colormap, and an image matrix. The colormap of an indexed image is an ordered set of values that represent the colors in the image. The size of the colormap matrix is n -by-3 for an indexed image containing n different colors. Each row in the colormap matrix represents a 1 x 3 red, green, blue (RGB) color vector which specifies the respective intensities of these three prime colors. These three colors add to produce any color in the spectrum. The intensities of the red, green, and blue components range from 0.0 (black) to 1.0 (full intensity). The elements in the image matrix refer to the corresponding rows in the colormap matrix. The colormap matrix adds the intensities of the three prime colors to produce the desired color effect at the corresponding spatial location in the color image. Since the three prime colors can produce any color in the spectrum, a grayscale image can also be represented by an indexed digital image having a colormap producing the desired levels of gray between black and white. Most indexed images have an 8-bit resolution, thus producing 256 unique colors.

2.1.4 RGB Image

This image is one of the standard ways to represent high resolution color data. The red, green, and blue (RGB) components of an RGB image are stored by computers as three separate intensity matrices. Each of the three matrices have the same row and column dimensions as the original image. The indices of corresponding elements from each of the three color intensity matrices combine to create the actual pixel color at the given spatial coordinates in the RGB image. The resolution of each color intensity matrix is 8-bit, thus resulting in a RGB image resolution of 24-bits. As a result, RGB images can produce up to 16,777,216 unique colors. RGB images are produced by most of the standard color scanners in the market today. These types of images are useful to perform several image processing operations on color data. An indexed image, for example, could be transformed into an RGB image before performing some image enhancement operations (e.g. filtering). These operations could then be applied to each of the three color intensity matrices separately. If this pre-conversion is not performed, the image processing operations would be carried out on the indices in the indexed image matrix, producing incorrect results.

2.2 Image Enhancement and Restoration

One of the most commonly used techniques in traditional image processing is image enhancement. Almost all image processing tasks involve some sort of enhancement, either as a desired end result or as a preprocess to some other image processing operation. The main objective of image enhancement techniques is to process a given image so that the result is more suitable and useful than the original image for a particular application. These techniques can be used to improve an image's contrast and brightness characteristics, reduce its noise content, or sharpen its details. However, what

is considered an improvement in an image is often subjective and dependent upon the application, and the judgment of the observer [Baxes, 1994]. When images are enhanced for human viewers the objective may be to improve the perceptual aspects of the image (e.g. contrast, brightness). Whereas in other applications (e.g. object identification by machines) an image may be preprocessed to aid machine performance. As a result, this processed image may not appeal to the human observer. Thus, one application's enhancement may be another's degradation.

Because the goal of image enhancement is dependent on the context of the application, these techniques can be grouped as either *subjective enhancements* or *objective enhancements*. Subjective enhancements are used to make an image more visually appealing to the observer. As a result, subjective enhancement techniques are usually ad hoc and applied with user discretion repeatedly until the observer feels that the processed image yields the details necessary for the particular application. Objective enhancements, on the other hand, are used to correct images for known degradations. These techniques are not necessarily used to make the processed image more appealing to the observer. Instead they are used to correct known faults based on the measurements from the original system. The enhancement is carefully computed from these measurements and applied once to correct the degraded image back to its mathematically correct form, regardless of its appearance. Objective image enhancement techniques cross over into the image restoration class of image processing. All image restoration techniques are based on known, measured, or accurately surmised degradations of the original image [Baxes, 1994]. As a result, image restoration techniques are ideal for restoring images with problems like improper focus, repetitive noise, geometric distortion, motion blur etc. Sometimes the degradation induced into an image may not be known. In this instance the image would hold several clues to the factors that led to its degradation. By investigating these clues an estimate of the degradation may be determined and used to inversely filter the image to reverse the process of the original degradation.

Over the past 30 years image enhancement and restoration have been the subjects of extensive research and a large number of publications have appeared on these topics during this period. However, only a few of these techniques are actually applicable in realistic circumstances of limited computational time and costs, workable image sizes, and insufficient availability of *a priori* information regarding image degradation. This section introduces some of the relevant enhancement and restoration techniques that may be useful in achieving the main objectives of this research.

2.2.1 Image Histogram and Histogram Equalization

An image histogram is a chart that shows the distribution of intensities or colors in an image. Most of the contrast enhancement techniques of image processing are related to somehow manipulating the image histogram. In a grayscale image the image histogram can be shown as a distribution graph of the number of pixels that fall into the various gray level steps. Image histograms in color images can be represented as a range of pixels that fall into each color bin. As far as this chapter is concerned all the image processing operations related to image histograms are illustrated with the help of grayscale images.

The image histogram allows an easy first interpretation of the concentration of pixels versus the brightness of an image. As a result, image histograms can be used to quantitatively identify low or high contrast images, as well as the amount of the dynamic range used by an image. An image having a low contrast shows a tightly grouped mound of pixels occupying a small range of the available gray levels. A high contrast image appears as a bimodal (two peaks at the outer brightness regions) distribution in an image histogram, leaving the middle gray levels unoccupied. The actual dynamic range of an image is the range of gray levels occupied by its image histogram. A low dynamic range implies low contrast, whereas a large dynamic range illustrates a well-balanced image (except high contrast images).

Histogram equalization techniques are used to manipulate the image histogram in several ways in order to enhance the contrast of an image. As mentioned earlier, the criteria for judging a processed image's contrast as good or bad are based entirely on the image's intended application. Histogram equalization is a *pixel point process* which involves the addition, subtraction, multiplication, or division of a constant value to every pixel of an image. There is no emphasis given to the spatial attributes of the image when a point operation like histogram equalization is performed on an image. This is carried out with the use of a mapping function which maps the brightness of the pixels in the original image to another value resulting in an enhanced image. The mapping function is chosen depending upon the desired modifications in the contrast of an image. Usually the goal of histogram equalization is to redistribute the image histogram so that it occupies the full dynamic range of the available gray levels, and at the same time is well balanced (i.e. flat). The cumulative histogram of the processed image in this case is approximately linear. In order to make some other attribute more clear sometimes it is desirable to produce an image which does not have well-balanced contrast attributes. In this case histogram equalization techniques may be used to create a mapping function that would result in an image histogram having the desired characteristics. The detailed process of choosing different types of mapping functions which produce various contrast enhancements to a degraded image can be found in the image processing literature [Pratt, 1978; Gonzalez et. al., 1987; Ekstrom, 1984; Jähne, 1993]. An example of an image histogram equalization technique is illustrated in figure 2.2 in which the contrast of an 8-bit grayscale image is enhanced. This operation is performed by flattening the original image histogram, thus redistributing the concentrated intensities throughout the full intensity range of the image. Upon performing the equalization, the resulting image has sharper contrast which makes it easier to discern certain details. However, it is evident that the enhanced image looks more unnatural (due to its higher contrast) as compared to the original.

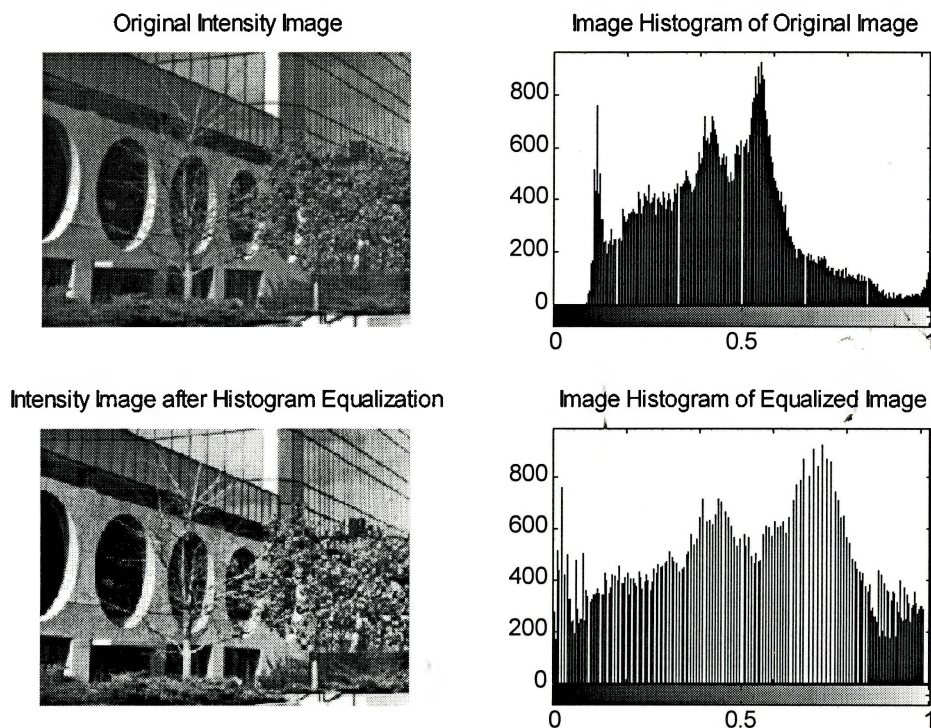


Figure 2.2 Use of image histogram and histogram equalization techniques to enhance the contrast of an 8-bit grayscale image

2.2.2 Image Adjustment

Another set of tools for enhancing the contrast of a degraded image is classified as image adjustment. These tools can compensate for flawed image histograms using three main types of adjustments. They are *input cropping*, *output cropping*, and *gamma correction*. These techniques can be used for both grayscale and color images. Image adjustment can stretch, or brighten the intensity ranges of grayscale images by mapping a certain intensity range from the original image on to a desired output intensity range. In the case of color images, these operations are performed separately on the red, green, and blue intensity components.

Input cropping, or intensity stretching, is responsible for mapping a certain section of the original image intensities to a full output intensity range. As a result, this operation

is useful for clearly displaying features concentrated in a particular part of an image's intensity range by mapping this range onto the full color range of your display. Another common application of input cropping is to map nearly black or nearly white colors in a grayscale image to black or white, respectively. This is useful to prepare scanned grayscale images for conversion to binary images. The images in figure 2.3 show the result of mapping the intensities between 0.05 and 0.6 to the full intensity range (i.e. 0.0 to 1.0). The accompanying image histograms give an indication of the modifications performed by this operation [Thompson et. al., 1993].

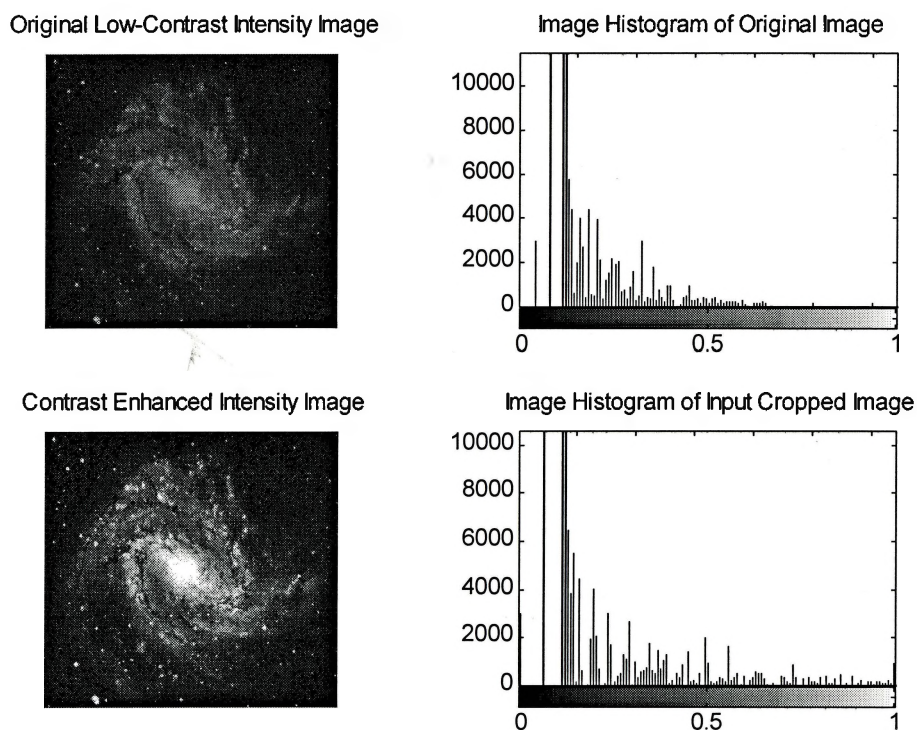


Figure 2.3 Use of image adjustment (input cropping) to enhance the contrast of a degraded intensity (grayscale) image

Output cropping is the complement operation to input cropping. This technique is responsible for restricting the intensity range of the output image. This operation is

mainly used to lighten or darken an image, or map the black parts of an image to 50% gray.

The final image adjustment operation is gamma correction which is also an intensity mapping function. However, in this case the input intensity value is mapped onto another value using the following power-based function:

$$y = x^\gamma \quad 2.1$$

where x is an input intensity, and y is the resulting output intensity value depending upon the power γ .

2.2.3 Filtering

Most images are affected to some extent by noise, which is unexplained variation in visual data. It would be a lot simpler to analyze images after this noise has been filtered out. Filters provide an aid to visual interpretation of images by either enhancing or restoring images. As a result, filtering operations are generally used as precursors to further image processing techniques like image analysis. Most of the contrast enhancement operations are classified as *pixel point operations* since they manipulate each pixel in the input image individually, without any consideration for spatial attributes of the image. However, spatial filtering is a *pixel group operation* which works on a group of input pixels surrounding a center pixel. The neighboring pixels provide valuable information about the brightness trends in the area being processed. A neighborhood where rapid brightness transitions are prevalent indicates a high spatial frequency, whereas a low spatial frequency is indicated by slowly changing brightness transitions. The highest frequencies in an image are found wherever sharp edges or points are present. Spatial frequencies in an image may be oriented horizontally, vertically, or at any diagonal.

An image can be filtered to accentuate or remove a band of spatial frequencies, such as low or high frequencies. These filtering operations are called spatial filtering operations. If the image is passed through a frequency transform it can be decomposed

from its spatial-domain form of brightness, into a frequency-domain form of fundamental frequency components. Each frequency component has a magnitude and phase value [Baxes, 1994]. The frequency form of an image can also be depicted as an image. Identical filtering operations, compared to spatial filtering, can also be carried out on the frequency domain image.

In this subsection both filtering methods (spatial & frequency domain) will be considered in further detail. The advantages of using each method are illustrated with the help of several types of filters, which are a part of these methods. Image filtering serves the dual purposes of image enhancement as well as image restoration. Some filters (e.g. edge detection filters) are responsible for image enhancement, whereas others (e.g. moving average filter) are responsible for restoring images by removing noise and averaging over nearby points.

2.2.3.1 Filtering in Spatial Domain

As mentioned above, filters in the spatial domain accentuate low or high spatial frequencies of an image for the purposes of image smoothing and edge detection, respectively. A subset of spatial filtering operations is responsible for highlighting only the sharp transitions of an image, such as the edges of objects within images. These operations are called edge enhancement operations.

Spatial linear filters are implemented through a process called spatial convolution, which is responsible for calculating the variation in pixel brightnesses within the neighborhood of the pixel being processed. Convolution is a mathematically complex method mainly used in signal processing. The spatial convolution method is also called finite impulse response (FIR) filtering. Majority of the spatial filters used in image processing are a 2-Dimensional extension of FIR filters. These filters are mainly used because of their zero phase property when filtering an image, thus maintaining the image's integrity once processed. As a result, a 2D-FIR filtered image doesn't look distorted when compared to its original form.

As in pixel point processing, the spatial convolution process moves across the image pixel by pixel, with each output pixel's brightness dependent on the neighboring pixels surrounding the pixel being processed. The spatial frequency activity in the neighborhood of the pixel being processed is calculated by the spatial convolution and filtering is performed based on this information. A weighted average of brightness values of the input pixel and its neighboring pixels is used to calculate the resulting output pixel brightness. The group of pixels used as a neighborhood in the weighted average calculation is called the kernel. Generally, the dimensions of the kernel are that of a square, with an odd number of weights in each dimension. The larger the size of a kernel, the more is the flexibility and precision of the spatial filter due to the larger size of the neighborhood accounted for in the calculation. A spatial convolution carried out with a 3 x 3 kernel requires 9 convolution coefficients which can be represented as an ordered array, as follows:

$$h = \begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix} \quad 2.2$$

This array of coefficients is called the convolution mask. In the above case the coefficient $w_{0,0}$ is the weight applied to the input pixel having a neighborhood of eight pixels. When a convolution mask is placed over an input pixel, the pixel and its eight neighbors are multiplied by their respective convolution coefficients, and the multiplicands are summed [Baxes, 1994]. The result is placed in the output image at the same spatial location as the input pixel. This process is repeated until each input pixel is convoluted to produce a spatially filtered image. The spatial convolution process of filtering images can be thought of as computing discrete equations in the following form:

$$B_{i,j} = \left\{ \begin{array}{lll} w_{-1,-1}A_{i-1,j-1} & + w_{-1,0}A_{i-1,j} & + w_{-1,1}A_{i-1,j+1} \\ + w_{0,-1}A_{i,j-1} & + w_{0,0}A_{i,j} & + w_{0,1}A_{i,j+1} \\ + w_{1,-1}A_{i+1,j-1} & + w_{1,0}A_{i+1,j} & + w_{1,1}A_{i+1,j+1} \end{array} \right\} \quad 2.3$$

where; $B_{i, j}$ is the output pixel's brightness value at the same spatial location (i, j) after filtering the input pixel $A_{i, j}$ using a 3×3 spatial convolution mask.

Various effects of a spatial linear filter can be obtained by assigning different weights to the convolution mask. Depending upon these weights, certain pixels in the convolution kernel will have more or less influence on the weighted average. If all the weights in the convolution mask are positive then the effect of the filter is to smooth the image. However, if some of the weights are negative then the filter outputs a weighted difference between pixel values. This has the effect of emphasizing edges of objects within an image.

Filters using a weighted average are of the linear kind since the calculations involve the summation of brightnesses multiplied by weights of a kernel. Non-linear spatial filters are potentially more powerful than linear filters due to their ability to reduce noise levels without simultaneously smoothing edges. As a result, these types of filters are mainly used for image restoration purposes. However, they too use a pixel group process to operate on an input pixel by using a kernel of surrounding input pixels. Unlike almost all linear filters, non-linear filters operate with different principles to combine the group's input brightnesses, thus making it quite difficult to categorize them.

Moving Average Smoothing Filter

The moving average or box filter is the simplest of the smoothing linear filters in the spatial domain. This filter is a spatial low-pass filter which has the effect of passing, or leaving untouched, the low spatial frequency components of an image. It replaces each pixel brightness by the average of the neighboring pixel brightnesses in the convolution kernel. All low-pass filters have positive coefficients in the convolution mask which sum to 1. The output pixel brightness value after applying a moving average filter in a region of approximately constant brightness pixels is the same as the average input pixel brightness. However, input regions having a high spatial frequency in pixel brightnesses produce output pixel regions with minimal transitions in brightness values. As a result,

the visual effect of using moving average filters is image blurring. This is illustrated in figure 2.4 which shows the use of a moving average filter with a 3 x 3 convolution mask on an 8-bit intensity image. By subtracting a moving average filtered image from its original and adding the result to the original image, a sharpened image may be created for enhancement purposes. This operation is called *unsharp masking* enhancement.

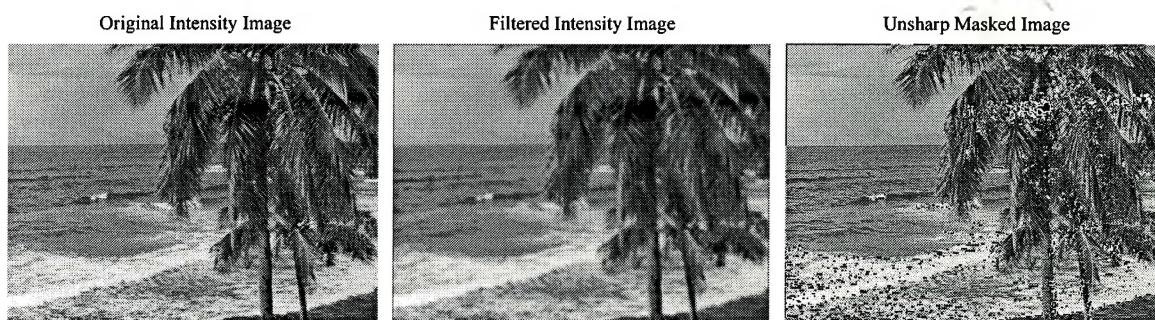


Figure 2.4 Use of a 3 x 3 moving average filter for smoothing and unsharp masking enhancement

Although the moving average smoothing filter is simple and computationally less time consuming than other filters, it suffers from some drawbacks. First, this filter is not isotropic (i.e. circularly symmetric), thus the smoothing is not consistent in all directions. As a result, this filter smoothes more along diagonals as compared to the rows and columns of an image. Second, discontinuities in the smoothed image exist due to the weights in the convolution mask being cut-off abruptly rather than decaying gradually to zero [Glasbey et. al., 1995].

Gaussian Low-Pass Smoothing Filter

Gaussian low-pass filters are the only ones that are approximately rotationally symmetric. They also overcome the discontinuity drawbacks of moving average filters because the weights in their convolution mask decay to zero. Due to their separable form these filters can be applied in two steps. First in a column operation using the column

weights of the convolution mask, followed by a row operation using the row weights, or vice versa. As a result, these filters require far less computational time compared to the ones that are not separable. Gaussian low-pass filters have weights in the convolution mask which are specified by the probability density function of a bivariate Gaussian distribution with a user specified variance σ^2 :

$$w_{ij} = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{(i^2 + j^2)}{2\sigma^2}\right\} \quad \text{for } i, j = -[3\sigma], \dots, [3\sigma] \quad 2.4$$

The divisor of $2\pi\sigma^2$ ensures that the weights of the convolution mask of a Gaussian low-pass filter sum to unity (approximately) which is a common convention of smoothing filters. Figure 2.5 illustrates the use of a Gaussian low-pass smoothing filter with $\sigma^2 = 0.49$ and a 3 x 3 convolution mask on an 8-bit intensity image.



Figure 2.5 Gaussian low-pass filtering ($\sigma^2 = 0.49$) with 3 x 3 convolution mask to smooth an 8-bit intensity image

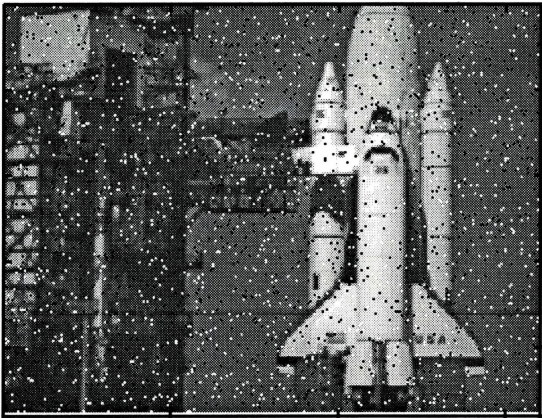
As compared to figure 2.4 it can be seen that the Gaussian filtered image is smoother with no discontinuities. This is clearly evident by comparing the unsharp mask enhanced images using both smoothing filters. The unsharp mask enhancement process is carried out by adding the difference between the original and the smoothed images to the original image. Thus, if the unsharp masked image shows more crispness (figure 2.4), it can be concluded that the smoothing performed by that filter displayed more

discontinuities. As a result, the Gaussian low-pass filter performs a more suitable filtering as compared to the moving average low-pass filter.

Moving Median Nonlinear Filter

Since both noise and edges are high spatial frequency components of an image, linear smoothing filters inevitably blur edges while removing noise. Nonlinear filters however, are able to simultaneously reduce noise and preserve edges. The simplest and most widely used nonlinear spatial filter is the moving median filter. This filter is similar to the moving average filter, except that it produces the median, rather than the mean, brightness value at the output pixel from the group of surrounding input pixel brightnesses. The moving median filter works by evaluating the pixel brightnesses in the convolution kernel and determining which pixel brightness value from the group is the median. This is done by placing all the pixel brightness values in the kernel in ascending order and choosing the middle value. Bright noise spikes in an input image are efficiently removed by the median filter since brighter values tend to be at the top of the ascending order of pixel brightnesses. As a result, noise caused by the presence of outlier pixels (i.e. extreme pixel brightnesses) is removed by median filters. This type of noise is commonly called '*salt & pepper noise*' and it originates during discretization or scanning of images. Figure 2.6 illustrates the use of median filtering to remove '*salt & pepper noise*' from an intensity image. A convolution kernel with a 3 x 3 dimension was used to filter the degraded image.

Shuttle Image Before Median Filtering



Shuttle Image After Median Filtering

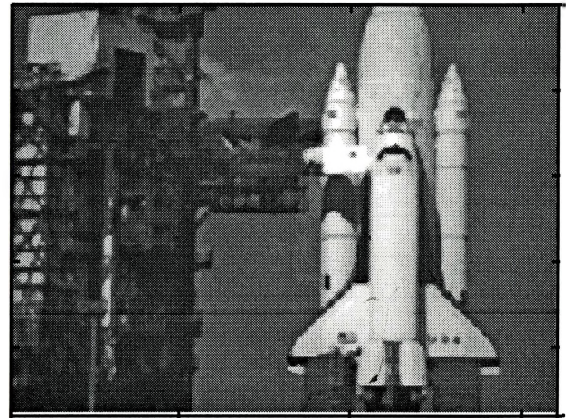


Figure 2.6 Use of a moving median filter to remove outlier pixels from an 8-bit intensity image

The moving median filter is a nonlinear filter, thus it reduces noise simultaneously while preserving the edges in the image. However, it is impossible for the median filter to preserve extremely sharp details within an image since noise removal requires some smoothing of high frequency components. This fact is illustrated in figure 2.7 which shows the use of a 3 x 3 median filter to remove sensor induced periodic noise from an 8-bit intensity image of a streak on photographic film. As seen, the median filter removes most of the noise, however the streak itself becomes blurry since the filter is not able to preserve the high frequency component present in it.

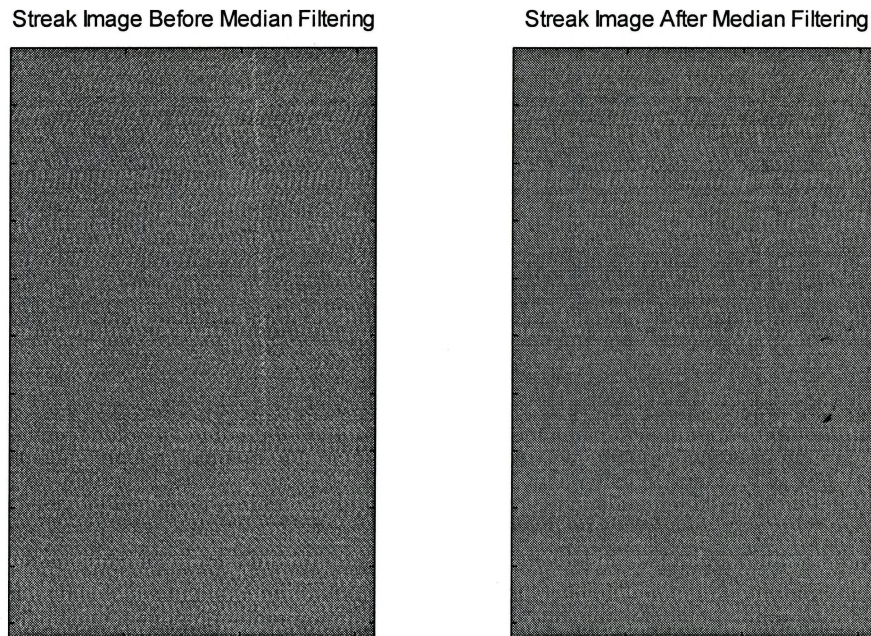


Figure 2.7 Use of a moving median filter to remove sensor induced periodic noise from an 8-bit intensity image

Wiener Low-Pass Adaptive Filter

Out of the linear spatial filters which smooth images for reduction of noise the Wiener filter produces the best results. If the amount of noise and blurring in an image are known, then the Wiener filter can be used to optimally restore the original image. Due to its adaptive nature the Wiener filter is more selective than other linear filters in preserving edges and other high frequency components of an image. Wiener filters work best when the noise is constant power additive noise, like Gaussian white noise. The Wiener filter uses a pixel-wise adaptive Wiener method based on statistics estimated from a local neighborhood of each input pixel. This filter tailors itself to the local variation in image brightness by estimating both the input neighborhood pixel brightness mean and variance. If the variation in brightness values is large, the Wiener filter performs little smoothing; however, it performs more smoothing if the variation is low. The coefficients

of the convolution mask in a Wiener adaptive filter are calculated using the following equation:

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (a(n_1, n_2) - \mu) \quad 2.5$$

where μ and σ^2 are the local pixel brightness mean and variance, respectively; b is the convolution mask coefficient; a is the input pixel brightness; and v^2 is the noise variance [Thompson et. al., 1993].

Since the Wiener filter has to perform new calculations for every input pixel neighborhood, the implementation of this filter is both computationally intensive and time consuming. The use of a 3 x 3 convolution masked Wiener filter on the intensity image of a streak on photographic film is illustrated in figure 2.8. Upon comparing the filtered image with that of figure 2.7 it is observable that the Wiener filter outperforms the moving median filter in this particular case. However, other nonlinear filters have been proven to outperform all linear filters including the Wiener filter [Glasbey et. al., 1995].

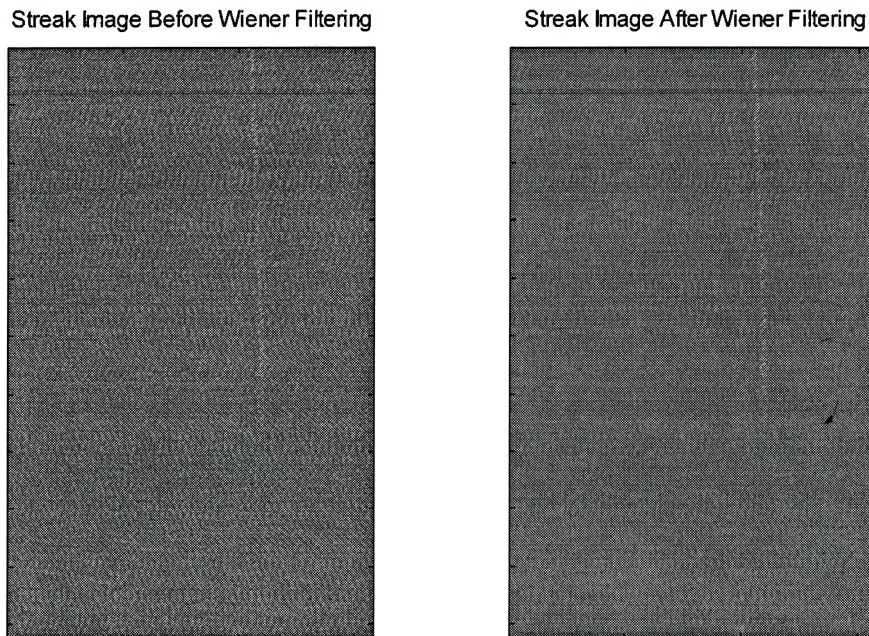


Figure 2.8 Use of a Wiener filter with a convolution mask of dimension 3×3 to both smooth noise and preserve edges in an 8-bit intensity image

Further details regarding the exact algorithm used by the adaptive Wiener filter may be gathered from several texts on statistics and digital image processing [Glasbey et. al., 1995; Thompson et. al., 1993; Gonzalez et. al., 1987; Ekstrom, 1984].

High-Pass Sharpening Filter

The high-pass filter has the opposite effect of the low-pass filter. This filter accentuates high frequency components of an image while leaving the low frequency components untouched. Common high-pass filter convolution masks contain a high positive coefficient value in the center location surrounded by small negative coefficient values such that the sum of all the coefficients is unity. This means that the center pixel in the group of input pixels being filtered carries a high influence whereas the surrounding pixels act to oppose it. If the brightness value of the center pixel is vastly different than the surrounding brightnesses (i.e. an edge is present), the surrounding pixel

effect becomes negligible, and the output value of the center pixel is accentuated. However, if the brightness values of all the input pixels in the neighborhood are approximately equal (i.e. no edge is present), then the output value is still based on weighted difference, but this difference is small. This means that the high-pass filter doesn't attenuate the low frequency spatial components of an image, rather it emphasizes the high frequency components only. As a result, the image looks coarser after high-pass filtering is performed. The use of a high-pass filter with a 3 x 3 convolution mask on an intensity image is illustrated in figure 2.9. Note that all the details of the waves and ground that were obscured in the original image are very apparent after filtering.

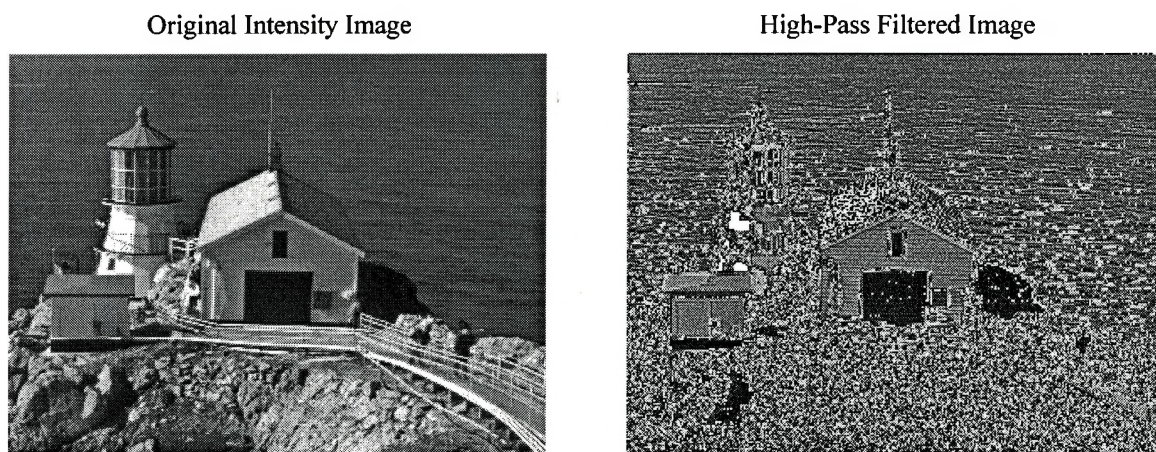


Figure 2.9 Use of a 3 x 3 high-pass spatial filter to sharpen the appearance of an 8-bit intensity image

First-Derivative Edge Detection Filters

Edge detection or enhancement operations on images show only edge details of an image once it is processed. These edges appear as object outlines in the image which can be used for the purposes of subsequent image analysis operations like feature or object recognition. As mentioned earlier, to emphasize edges in an image some of the coefficients in the convolution mask of a spatial filter have to be negative and the mask

coefficients should sum to zero. As a result, these filters produce as output the differences in pixel values in the input image. The output magnitude of the pixel brightness values will be large (positive and negative) if there is a large difference in adjacent input pixel brightnesses. The following convolution mask is an example of a first derivative vertical edge detection filter:

$$h_v = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad 2.6$$

This choice of coefficients estimates the first derivative of the image intensity along a row of an image upon taking the weighted average of the input neighborhood intensities around the input pixel [Glasbey et. al., 1995]. The convolution mask for a first derivative horizontal edge detection filter is simply the negative transpose of the vertical edge detection filter [i.e. $h_h = (-h_v)^T$]. Figure 2.10 illustrates the use of both first-derivative vertical and horizontal edge detecting filters on an intensity image.

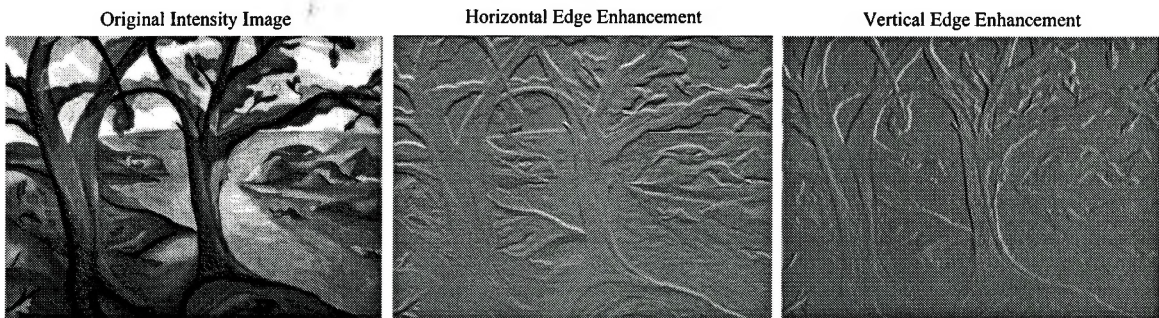


Figure 2.10 Use of first-derivative vertical and horizontal edge detection filters to enhance an 8-bit intensity image

Since first-derivative filters produce output brightnesses with both positive and negative magnitudes, these images have been displayed with zero pixel brightness as mid-gray, positive values as lighter grays and negative values as darker grays. This results in an embossed effect and the resulting image is perceived as 3D, even though it isn't.

Second-Derivative Edge Detection Filters

Second-derivative filters, unlike first-derivative ones, can be isotropic, and therefore responsive to edges in any direction while remaining linear [Glasbey et. al., 1995]. The Laplacian edge enhancement filter is an omnidirectional filter that highlights edges in all directions of an image, regardless of their orientation. The common Laplacian filter convolution mask is composed of a high positive coefficient in the center location and small negative coefficients surrounding it such that the sum of all coefficients is zero. The following is a common Laplacian convolution mask:

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad 2.7$$

It can be shown that with these coefficients the 2D-Laplacian operator, $\nabla^2 (= \partial^2/\partial x^2 + \partial^2/\partial y^2)$, of the input pixel brightness can be closely approximated. There is also a simpler version of a Laplacian filter which also approximates ∇^2 with the following convolution mask:

$$\nabla^2 \approx \frac{4}{(\alpha + 1)} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}; \text{ where } \alpha \text{ varies between } 0 \rightarrow 1 \quad 2.8$$

Figure 2.11 illustrates the use of the latter of the two convolution masks (dimension 3 x 3) of a Laplacian filter with an α value of 0 to detect the edges of an intensity image. As before, the Laplacian filtered image is shown with an embossed effect.

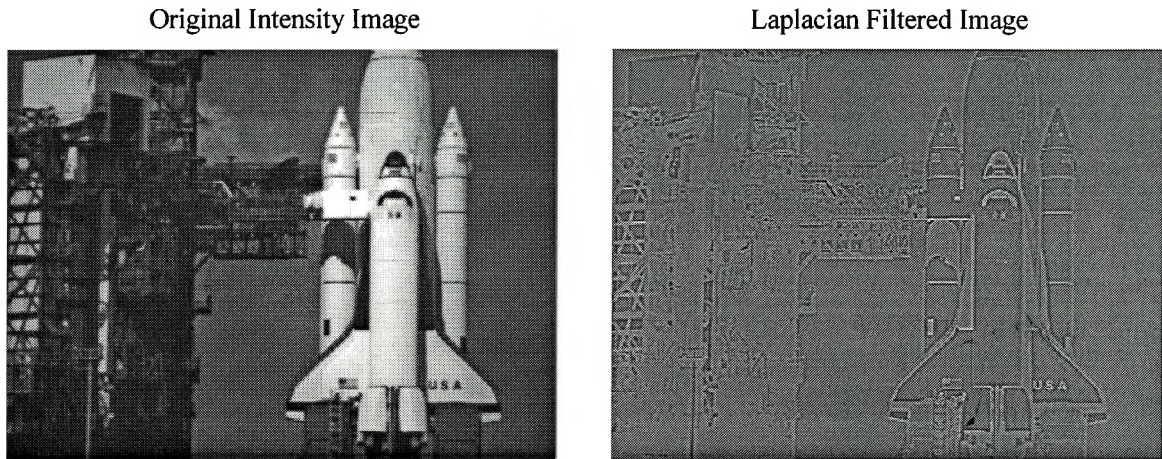


Figure 2.11 Use of a 3 x 3 Laplacian filter ($\alpha = 0$) to detect edges in all directions of an 8-bit intensity image

Since the detection of edges means the accentuation of high frequency components of an image, Laplacian filters also accentuate noise along with the edges of an image. To reduce the effects of noise, the above Laplacian filter may be applied to an image produced after Gaussian smoothing of the original image. Such filters are also called Laplacian-of-Gaussian filters. Since the Gaussian filter is a low-pass smoothing filter it is important to choose an appropriate variance for the calculation of the coefficients in its convolution mask. If the variance is too large the number of apparent edges after applying the Laplacian-of-Gaussian filter would decrease. For further descriptions of the Laplacian-of-Gaussian and various other linear and nonlinear spatial filters, the reader may refer to one of several texts in the image processing literature [Pratt, 1978; Gonzalez et. al., 1987; Baxes, 1994, Glasbey et. al., 1995; Ekstrom, 1984; Jähne, 1993].

2.2.3.2 Filtering in Frequency Domain

Details within an image are comprised of low and high spatial frequency components which are dependent on the rates at which details transition from dark to

bright, and back to dark. These spatial frequencies have varying orientations within an image. A frequency transform converts an image from the spatial domain of pixel values to the frequency domain of fundamental frequency components. Each frequency component has a magnitude and phase value. As a result, instead of representing the image as an array of pixel brightnesses, the frequency transformed image is represented as an array of the magnitudes of spatial frequency components. This array can also be displayed as an intensity image where the pixel brightnesses represent the magnitudes of the various spatial frequency components at specified locations. In a frequency image the horizontal frequency is defined in the x direction whereas the vertical frequency is defined in the y direction. Diagonal frequencies are defined as points in the middle. Thus frequency transforms may be interpreted as a pictorial view of the various spatial frequencies of an image. All frequency transforms also have a corresponding inverse frequency transform which converts the image back to its spatial domain from the frequency domain. Several frequency transforms exist which convert spatial images into their corresponding frequency components and back. Some examples of frequency transforms are the Fourier, Cosine, Sine, and Karhunen-Loeve transforms. Each transform has certain unique characteristics since each transform uses a different frequency component (or basis) function to decompose an image [Baxes, 1994]. Out of these the most common transform is the Fourier transform. This transform converts a spatial image into a sum of many sine waves of different frequencies, amplitudes, and directions which can be considered the most fundamental frequency domain. The parameters specifying the sine waves are called Fourier coefficients. The type of Fourier transform used to process digital images is called the Discrete Fourier Transform (DFT). A faster and much more computationally efficient form of the DFT is the Fast Fourier Transform (FFT). Generally all digital image processing that requires Fourier transforms employs the FFT. This section will concentrate on the qualitative aspects of filtering using the FFT in image processing. Further theoretical and mathematical details

regarding the FFT and other digital image frequency transforms may be gathered from several image processing texts [Gonzalez et. al., 1987; Pratt, 1978; Glasbey et. al., 1995].

The Fourier transform of an image in the spatial domain is a two-dimensional process. First each row of pixels is processed, followed by each column. This results in another array of values, each having a magnitude and a phase part. Each element in this array represents a distinct spatial frequency component of the image. The frequency transform array has the same dimensions as the original image. The magnitude part of the frequency array can be viewed as an intensity image which shows the frequency components of the original images as illustrated in Figure 2.12. This figure illustrates a 2D-FFT image of an intensity image.

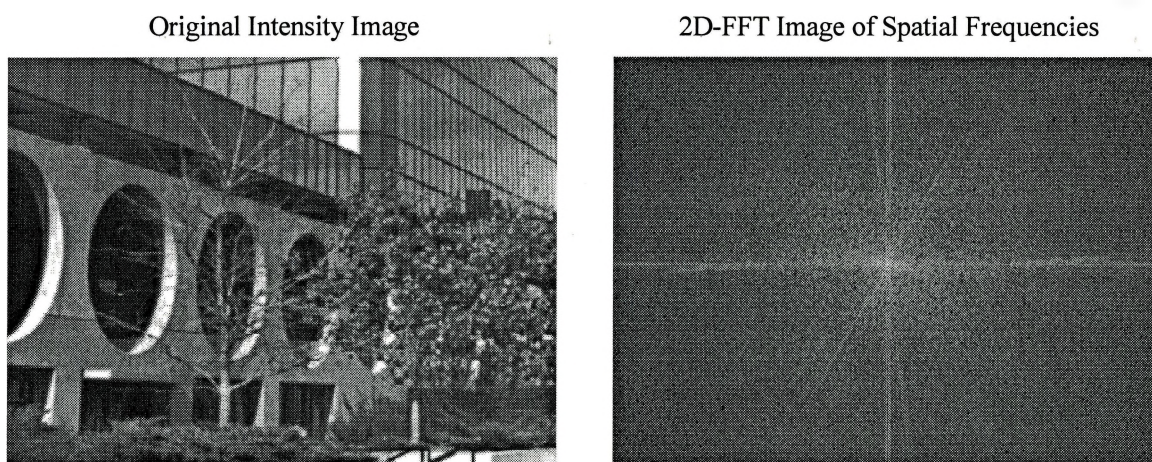


Figure 2.12 Use of 2D-FFT to display spatial frequencies and their magnitudes of an 8-bit intensity image

As seen in the above figure, the 2D-FFT image appears to be symmetric about its center. This is because the center is the zero-frequency point of the display. Horizontal and vertical axes exist through the center which define the horizontal and vertical frequencies, respectively. All frequencies begin at the center (i.e. at 0 or “DC” for direct current) and progress outward until the outermost frequency, which is the Nyquist

frequency ($1/2$ the sampling rate of the image). The magnitude of each frequency component is indicated by the brightness of the pixel at the location in the image [Baxes, 1994]. This image shows higher brightnesses near the lower frequency components (i.e. the axes) which indicates higher magnitudes in these regions. This is quite common in digital images since majority of the spatial information is gathered at lower frequencies. The magnitude image has been displayed on a logarithmic scale because otherwise all other pixels, except the ones near the center, would appear black compared to the large magnitudes (i.e. brightnesses) at low frequencies. The negative frequencies in the magnitude image are symmetric mirrors of the positive frequency quadrant, thus they convey redundant information. However, they are necessary for the inverse Fourier transform. Both magnitude and phase information is required to perform the inverse Fourier transform to convert the image back into the spatial domain.

Since the Fourier transform allows us to quickly view the spatial frequency content of an image, any repetitive noise patterns that might exist in the original image will clearly show in the 2D-FFT magnitude image. Figure 2.13 illustrates this point with the image of the streak on photographic film which has been corrupted by sensor induced periodic noise patterns.

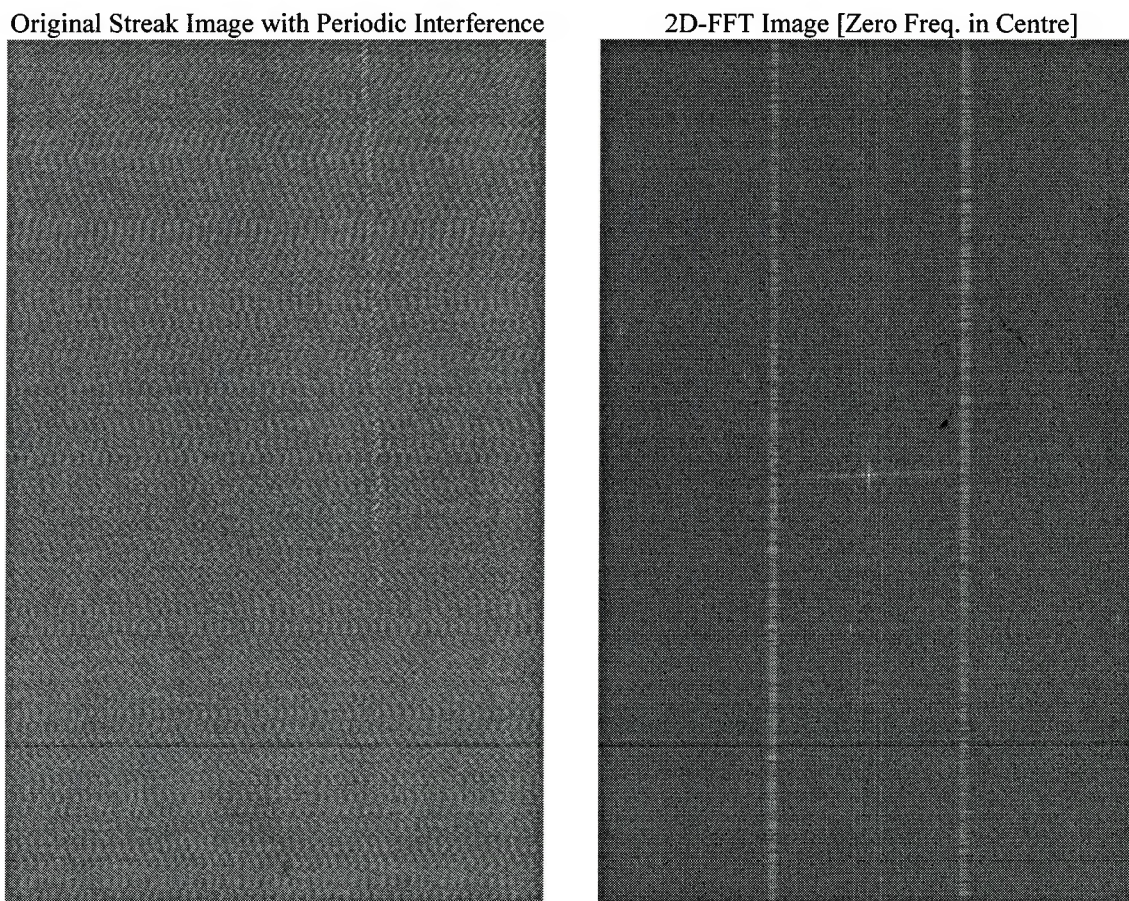


Figure 2.13 Use of 2D-FFT to identify the frequency component of sensor induced periodic noise patterns in an 8-bit intensity image

As seen in the original intensity image the noise patterns have a horizontal frequency component since the wavy pattern is in the vertical direction. Upon transforming this image in the frequency domain this noise pattern shows up clearly as an intensity burst on the horizontal axis in the magnitude image. Although the noise pattern has a horizontal frequency component, its frequency does vary. This results in the vertical elongation of the intensity burst in the frequency domain. Knowing this information about the frequency characteristics of the noise, several filters can be created

in the frequency domain to attenuate the noise patterns from the image. Once filtering is performed, the processed image can be inverse transformed back into the spatial domain.

Numerous filters were introduced in the previous sub-section which were used to perform filtering in the spatial domain using a convolution mask. Identical filtering operations are also performed on the magnitude image in the frequency domain. It has been theoretically proven [Gonzalez et. al., 1987] that spatial convolution between a spatial image and a convolution mask are identical to the multiplication of a frequency image and a frequency mask. However, in the frequency domain, filtering can be very selective. Thus, one can remove specific frequency components, or bands of frequency components, from the frequency image. Filtering in the frequency domain is performed by multiplying a frequency mask image by the transformed frequency image using a dual-image pixel point process. The frequency mask image has the same dimensions as the original (& frequency transformed) image and contains zeros and ones. A zero is in the locations wherever one wants to eliminate a frequency and a one everywhere else. As a result, wherever the frequency image is multiplied by a zero, it will become zero as well, and wherever it is multiplied by a one, it will remain unchanged. Once this filtering is performed, the processed frequency image is inverse Fourier transformed into the spatial domain. The resulting image will be devoid of the frequencies that were multiplied by zero. A frequency mask which uses this technique to eliminate specific frequencies or bands of frequencies from the original image is called a band-reject filter. However, it should be kept in mind that any spatial details having the same spatial frequencies will also be attenuated which might result in a subjective degradation of the image. Using the principle of the frequency mask, several filters may be created (e.g. low-pass, high-pass, band-pass filters) that remove the desired spatial frequencies by multiplying the appropriate locations of pixels in the magnitude image by zero. Values other than zero may also be used in frequency masks to control the degree of attenuation of corresponding frequencies. This leads to the idea of using two-dimensional windowing functions to create frequency masks for controlling the amount of attenuation

or accentuation of frequency components. The windowing function also helps overcome one of the shortcomings of Fourier transforms. The Fourier transform models the frequency representation of a spatial image as though it were periodic. Thus from the perspective of the Fourier transform, the image frame wraps around from its right edge to its left edge, and from its top edge to its bottom edge [Baxes, 1994]. If there are significant differences in the spatial details at the various edges of an image the Fourier transform perceives a discontinuity in the spatial information. Since spatial discontinuities contain a wide band of spatial frequency components the frequency image will contain erroneous frequency components. These erroneous components are a result of an attempt by the Fourier transform to model the spatial discontinuities in the image. This error becomes noticeable when the inverse Fourier transform is performed. The resulting image will have spatial distortion artifacts since the inverse Fourier transform tries to recreate the spatial discontinuities at the edges which really didn't exist in the original image. A windowing function may be used to match the brightness values of all edges before performing the Fourier transform. Most windowing functions smoothly reduce the pixel brightnesses of an image to zero as the edges are approached. Since all edges will now have values of zero, the perceived spatial discontinuity (by the Fourier transform) is now eliminated [Baxes, 1994]. Similar spatial discontinuities may also be eliminated from frequency mask images with the use of windowing functions. Instead of setting unwanted frequencies to zero a smooth transition to 0 may be used, thus reducing the effects of spatial distortion once the image is inverse transformed to the spatial domain. Several windowing functions may be used depending upon the rate at which pixel brightnesses need to approach 0. A triangular window linearly reduces pixel brightnesses from 1 to 0 as the edges of the image are approached. Other windowing functions include the Gaussian, Hamming, Hanning, etc. windows which perform smoother transitions that gradually change the pixel brightnesses from 1 to 0. Figure 2.14 illustrates the use of a frequency mask image with a band-reject filter to attenuate the sensor induced horizontal periodic noise patterns from the intensity image of the streak

on photographic film. After filtering, the inverse 2D-FFT intensity image is devoid of the periodic noise patterns, however, spatial details which had the same spatial frequency as the noise have also been reduced.

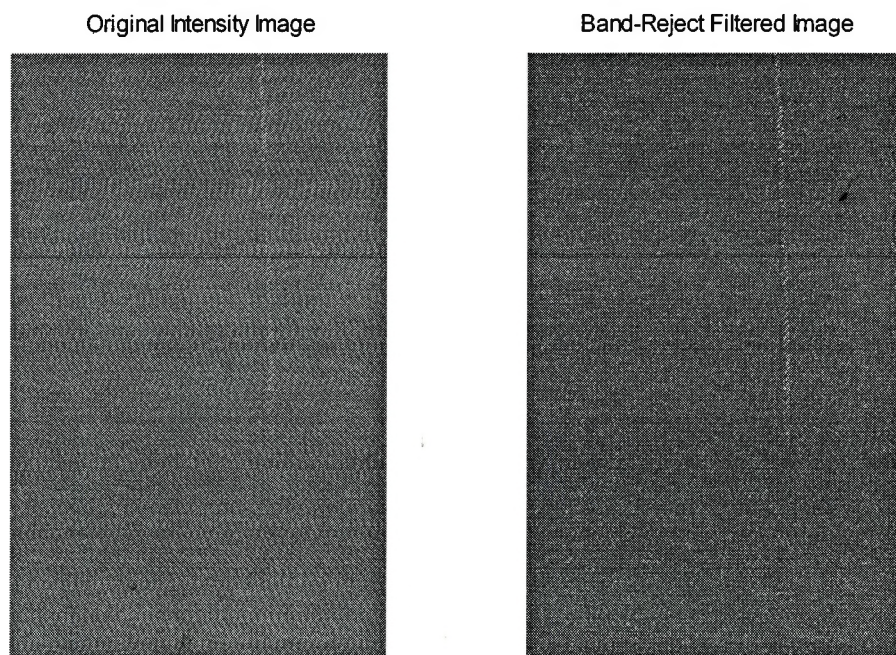


Figure 2.14 Use of a frequency domain band-reject filter to attenuate sensor induced noise frequency patterns from an 8-bit intensity image

The 2D-FFT magnitude image is also useful in comparing textural qualities of images on a common spectral base.

2.3 Image Analysis

A digital image exists as a large array of pixels with associated brightness values. However, interpretation of the scene contained in this native form of an image is left to the observer since the image contains no inherent intelligence about its contents. The main goal of image analysis operations is to produce an automated description of a given image. This is done by quantifying the elements of an image. The elements of interest are generally objects in an image (e.g. manufactured parts, blood cells, areas of land in a satellite image etc.). The quantification of these objects include measure of size, shape, color, texture etc. As a result, image analysis operations are largely used in applications that require the measurement and classification of image information. Image analysis operations are ideally suited for machine vision applications which can free human operators from the tedious, and sometimes inaccurate, manual inspection of objects of interest. Although the steps followed by image analysis operations are application specific, there is a general sequence in which most of these operations are performed.

The initial step in almost all image analysis processes is segmentation of the various parts (i.e. objects, regions, or features) of the image. This process is generally performed by thresholding the pixel brightness values in the original image at some threshold value. However, the initial results of segmentation do not always produce the desired results. Segmentation may yield a large set of parts which may touch or overlap. So it may be necessary to split these sets into further sub-parts or assemble them into various groups [Ekstrom, 1984]. This resegmentation of the parts obtained from the initial segmentation step is performed by morphological operations. These operations may also be used for classification of parts once they have been sufficiently segmented. Once the segmentation of objects is completed these parts may be measured and described in various types of geometric properties (number, shape, size, etc.) as well as in terms of their colors, texture, etc. It is important to remember that the above mentioned

sequence of image analysis operations assumes that the image has been preprocessed up to the stage where it is ready for segmentation. The preprocessing may include steps like image enhancement and restoration. Choosing appropriate preprocessing operations can help reduce image degradations, leading to better object detection and improved image analysis performance.

2.3.1 Segmentation

The first step in any image analysis operation is the simplification of the image into its basic regions or categories, which correspond to different objects or parts of objects. This is the domain of image segmentation operations. In a typical segmentation operation every pixel in the image is allocated to one of a number of different categories. It is important to perform segmentation in such a way that all the important features in the image are preserved. The definition of “important features” often depends on the requirements of a particular application. If image segmentation is done well then all other stages in image analysis are made simpler. There are three general approaches to image segmentation which are termed thresholding, edge-based segmentation, and region-based segmentation [Glasbey et. al., 1995]. In thresholding, pixels are allocated to categories depending upon the range of brightness values they possess and where they are categorized. In edge-based segmentation, pixels are classified as edge or non-edge depending upon the output of an edge filter applied to the image. Pixels that are not separated by an edge are allocated to the same category. Finally, region-based segmentation operates iteratively by grouping neighboring pixels that have a similar brightness value, and splitting groups of pixels that are dissimilar in brightness values [Glasbey et. al., 1995].

Out of these segmentation methods this sub-section concentrates on thresholding as the primary tool for classifying objects in an image. One can refer to several references at the end of this chapter for further details regarding image segmentation [Glasbey et. al., 1995; Gonzalez et. al., 1987; Ekstrom, 1984; Jähne, 1993; Pratt, 1978].

The simplest and most commonly used method of image segmentation is thresholding, in which each pixel is classified as light or dark depending on whether or not its brightness level exceeds a given threshold brightness value. Given a threshold t , the pixel located at location (i, j) in an image with a brightness value f_{ij} , is allocated to category 1 if $f_{ij} \leq t$. Otherwise, the pixel is allocated to category 2. Depending on the particular application, t is manually chosen by the observer after trying a range of different values and seeing which one works best at identifying the objects of interest. A threshold intensity value of $t = 0.50$ was chosen (after trying a range of values for t) in Figure 2.15 to segment an intensity image of bacterial growth on photographic film. All pixel brightnesses lesser than or equal to 0.50 were set to 0.0 (black) and the rest were set to 1.0 (white) thus creating a binary image after thresholding.

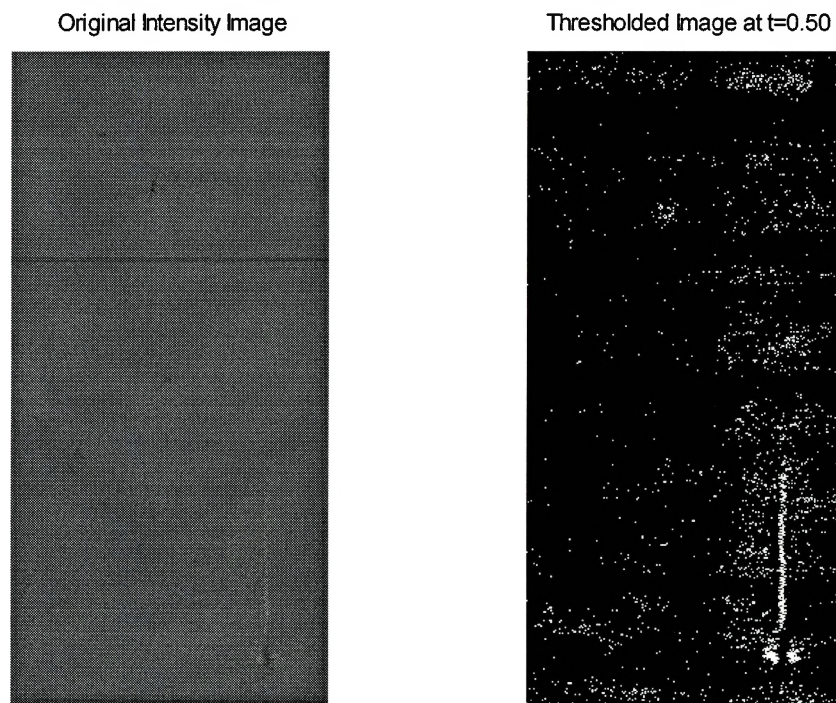


Figure 2.15 Thresholding used to create 2 categories at $t = 0.50$ on an 8-bit intensity image to segment bacterial growth on photographic film from the background

It can be seen that basic thresholding with 2 categories does an adequate job in categorizing the object of interest (bacterial growth) in the above image. However, more than one threshold value may be used if further categorization of an image is required. If two values of t are chosen, the resulting image will be segmented into three categories. This type of an approach is generally suited if there are more than two types of objects in a single image and they can be distinguished on the base of their brightnesses. If image analysis operations are being applied in an automated machine vision application, and the segmentation step requires thresholding an image, then t may either be set *a priori*, based on previous experience, or chosen automatically. Several algorithms are available that allow automatic thresholding by assigning t based on the brightness histogram of a particular image. The '*intermeans algorithm*' is one such technique for automatically choosing a single threshold [Glasbey et. al., 1995]. This algorithm begins at a specified guess of the possible threshold brightness value. From this value, the means of the pixel brightnesses, falling in the two created categories, are calculated. The threshold value is repositioned half way between the two means. Once again two new mean values are calculated and the threshold is repositioned half way. This is repeated until there is no further change in the threshold value between two successive iterations. Figure 2.16 illustrates the use of intermeans algorithm to choose a threshold value for segmentation of the bacterial growth on photographic film. The calculated threshold value in this case was 0.4734. Upon comparing figures 2.16 with 2.15 it can be seen that the intermeans algorithm does not *subjectively* perform as good as manually choosing the threshold brightness value.

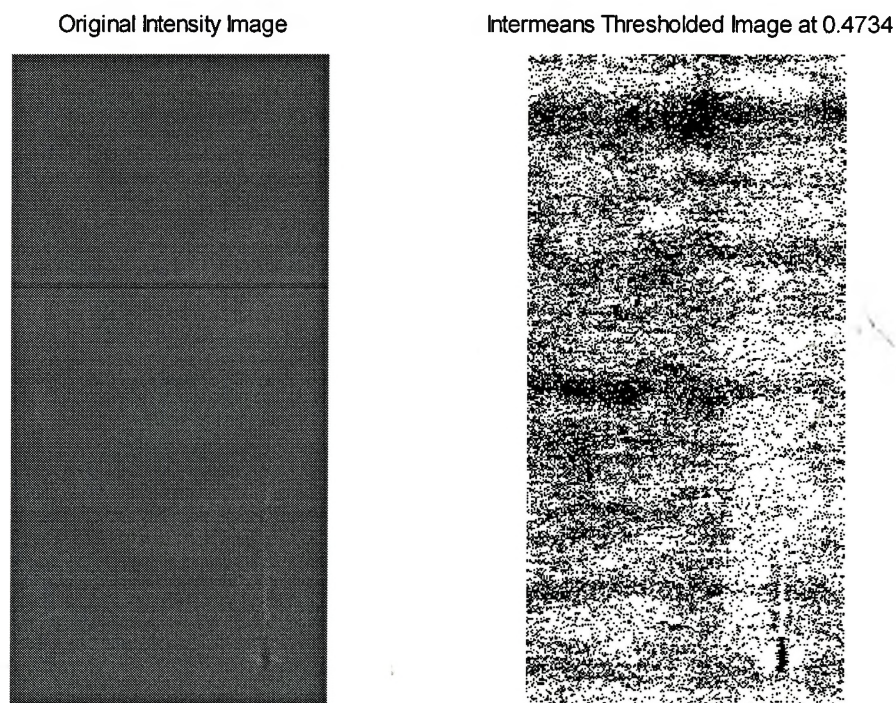


Figure 2.16 Use of the ‘intermeans algorithm’ to calculate threshold value $t = 0.4734$ for segmenting an 8-bit intensity image

Ideally, histogram based thresholding works best when there is good brightness separation between the various objects and the background of an image. If this is the case, then the brightness histogram would display a multi-modal distribution (one mode each for the objects and background) with few pixels between the modes. A threshold brightness value chosen between these modes would adequately separate the various objects and the background of the image. Unfortunately, this ideal situation rarely occurs in practice because of intrinsic variation in brightness values among pixels of the same type, and because of lighting variations in the image. In order to obtain better segmentation in such images it is necessary to take into account information regarding size and shape, along with brightness, of objects. A very powerful set of tools which help perform these tasks is provided by mathematical morphology.

2.3.2 Morphological Operations

Morphology is an approach to image analysis which is based on the assumption that an image consists of structures that may be handled by the use of set theory. Mathematical morphological operations are based on the ideas developed by J. Serra and G. Matheron of Ecole des Mines, France over the past 20 years [Glasbey et. al., 1995].

The study of the structure of objects within an image is defined as image morphology. After initial object discrimination with the help of various segmentation techniques, morphological operations may be used to clarify the underlying structure of the segmented objects. Morphological operations are primarily used for binary image processing, however, these ideas have also been extended to grayscale images. This sub-section describes the ideas of morphology first for binary images, and then examines the extension of these ideas to grayscale images.

Binary morphological operations establish the premises of image morphological processing. Grayscale morphological processing is simply a continuation of the binary techniques. Morphological operations work much like the spatial convolution pixel group operations. However, morphological operations logically combine input pixel brightnesses with a '*structuring element*' to look for specific patterns. Instead of multiplying pixel brightnesses with some weights and summing the result like the spatial convolution process, morphological operations use set theory notations, such as intersections (\cap), unions (\cup), complements (c), and subsets (\subset), to combine the input pixels logically into a resulting output pixel value [Baxes, 1994]. Within the context of a digital image, sets are groups of pixels, and the set terminology is a convenient way of describing what pixels lie in particular groups, which are considered distinct objects. It is assumed that the reader has some background in basic set theory before proceeding any further in this sub-section. If a reader is interested in acquiring a fuller knowledge of set theory (s)he may read one of many introductory books in this area e.g. [Vilenkin, 1968].

All binary morphological operations in an image are performed on its objects, which are represented by black pixels. Any white pixels in a binary image are considered the complement of black pixels. Like spatial convolution, the morphological operation process moves across the input image pixel by pixel, placing resulting pixels in the output image at the same spatial locations. The input pixels and its neighbors are logically compared at each location against a structuring element to determine the output pixel's logical value. The resulting output pixel value appears as either black (0) or white (1) in a binary image. The structuring element is an important tool in morphological operations. It can be thought of as a morphological mask, like the convolution mask, which consists of an array of logical values and is used to analyze the shape of an object. The structuring element is generally composed of square dimensions with each element taking on a value of 0 or 1, or X which is the "don't care" state [Baxes, 1994]. However, as already mentioned, black pixels in a binary image determine the shapes of objects, thus the actual shape of a structuring element is also defined by the locations of the black (0) elements within the array. These structuring elements can be placed on any pixel in the input image, although their rotation is not allowed while performing a particular morphological operation on an image. As a result, a reference pixel has to be defined within the array of the structuring element. The position of the reference pixel determines where the structuring element is placed. The choice of the shape of a structuring element (e.g. line, square, disc etc.) is determined by which features in the input image (e.g. linear, circular etc.) are to be suppressed or enhanced.

The mechanics of the binary morphological operations are similar to the spatial convolution process. Every pixel in the input image is evaluated with its neighboring pixels, where the size of the neighborhood is determined by the structuring element size, to determine a resulting output pixel value. The input pixel and its neighbors are logically compared to the values of the elements in the structuring element. Wherever a don't care state exists in the structuring element no comparison is made. In the case where all input pixels are identical to their corresponding structuring element values, the

resulting output pixel value is set to a predefined state (1 or 0). If one or more input pixels don't match their corresponding structuring element values then the output pixel value is set to its opposite state (i.e. 1 if it was 0, and vice versa). This process is repeated pixel by pixel for each pixel in the input image. This process is commonly called the '*Hit or Miss Transform*'. If the input pixel values match the corresponding structuring element values the evaluation is called a 'hit', otherwise it is a 'miss' [Baxes, 1994].

Erosion and Dilation

The two most basic morphological operations are those of erosion and dilation. Almost all morphological operations on images can be expressed as different combinations of erosion and dilation. The erosion operation uniformly removes pixels from the boundaries of objects, whereas the dilation operation, inverse of the erosion operation, adds pixels to the boundaries of objects. Both erosion and dilation operations are used to remove small-image object features, like noise spikes and ragged edges. As explained above, an object within a binary image is defined as a group of black (0) pixels and the background is defined as the white (1) pixels.

Knowing this convention, an example of a 3 x 3 square erosion morphological mask (structuring element) for the Hit or Miss Transform is as follows:

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}$$

where the output pixel value at location (x, y) is set to black (0) for a hit, otherwise it is set to white (1) for a miss.

There are two distinct cases to consider for fully understanding the effects of erosion with the above structuring element as it passes over image regions with various black and white pixel combinations. In each of the two cases, the morphological process compares each input pixel's brightness value against its corresponding mask value. When all nine pixels in the input neighborhood match their respective mask values a hit occurs and the

output pixel value is set to black (0). Even if one or more of the input pixel values don't match their respective mask values the output pixel value is set to white (1) [i.e. the output pixel becomes a part of the background]. When the input pixel is at the edge of an object in the original image, its input neighborhood would contain a mix of both black and white pixels. Since all of the input pixels will not be black, as in the case if the input pixel is in the middle of an object, the resulting output pixel at that location would become a part of the background.

Structuring elements can be of any predefined shape or size depending upon the requirements of the resulting effects it creates after erosion or dilation. If, for example, one needed a diagonal structuring element (NW to SE) of length 3 pixels, then one simply needs to define the above morphological mask with black (0) pixels in the diagonal locations of the 3 x 3 array with white (1) pixels in the remaining locations. Erosion and dilation with different shapes and sizes of structuring elements produce different results. Figure 2.17 illustrates the use of three shapes of structuring elements to erode a 2-bit binary image. It can be seen from this figure that the eroded objects show traits of their respective structuring elements.

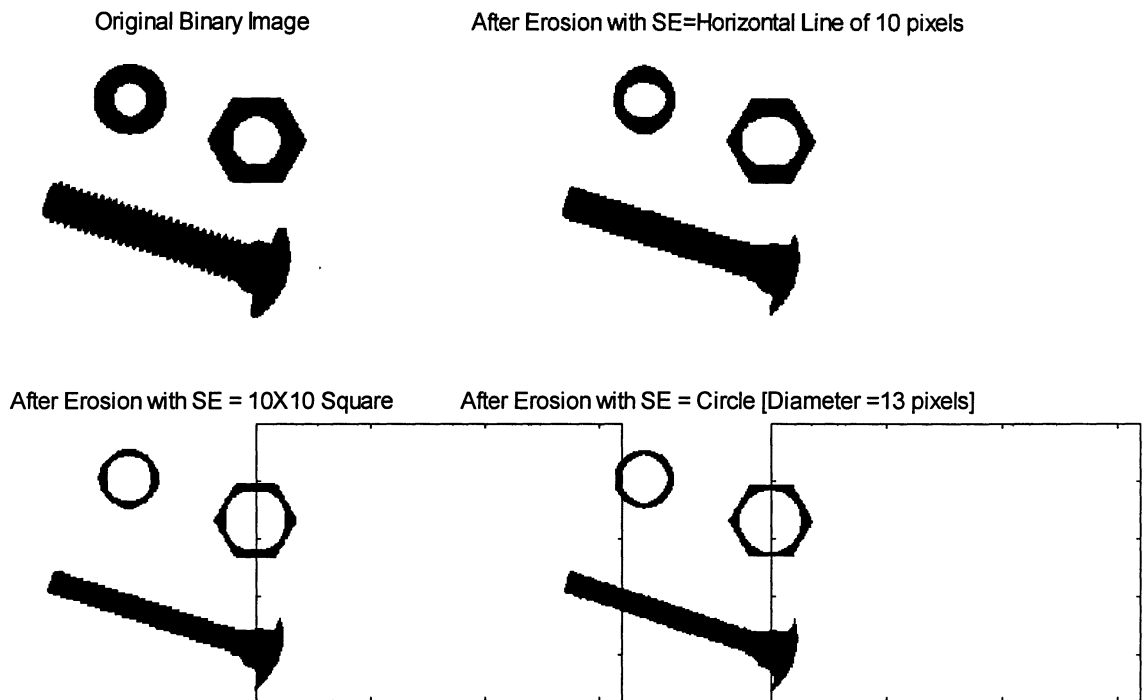


Figure 2.17 Examples of morphological erosion with various structuring elements, (a) Horizontal line, (b) Square, (c) Disc

The subjective effect of erosion using a given structuring element on an image is the same as if we tested whether the structuring element centered on each pixel of the image with the structuring element's reference pixel in turn, fits inside an object. If it does fit then the pixel is set to black, otherwise it's set to white. Any objects smaller than the structuring element are removed from the image, thus cleaning pixels resulting from noise spikes and ragged edges. In set theory notation, if A is a set (i.e. a binary image or part of it) and S is a structuring element with its reference pixel placed at (i, j) denoted by $S_{(i, j)}$, then the erosion of A by S (i.e. $A \ominus S$) is the set of all pixels for which S is contained within A [Glasbey et. al., 1995]:

$$A \ominus S = \{(i, j) : S_{(i, j)} \subset A\}.$$

Dilation of a binary image is the complementary operation to that of erosion. If the complement of a set A (i.e. a binary image or part of it) is denoted by A^c then the dilation of A by the structuring element S (i.e. $A \oplus S$) is defined as:

$$A \oplus S = (A^c \ominus S)^c. \quad 2.10$$

The subjective effect of dilation using a given structuring element on an image is the same as if we tested whether the structuring element centered on each pixel of the image with its reference pixel in turn, touches an object in the image. If it does touch then the pixel is set to black, otherwise it's set to white. This results in the enlargement of any objects in the original image. In terms of the Hit or Miss transform, morphological dilation is performed with a generalized 3 x 3 morphological mask as follows:

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

where the output pixel value at location (x,y) is set to white (1) for a hit, otherwise it is set to black (0) for a miss.

In this case only if all the input pixel values match their corresponding mask values (i.e. white) will the output pixel value be set to white (1). In every other case this value will be set to black (0). When the input pixel is near an object edge its neighborhood will contain a mix of black (0) and white (1) pixels. This would result in a miss, producing an output pixel value of 0. As a result, the dilation operation changes pixels which are in the background near object edges to be a part of the objects. Upon performing dilation with several shapes and sizes of structuring elements, it can be seen that the dilated objects show traits of the respective structuring element used. Dilations with the same three shapes of structuring elements on the same binary image as figure 2.17 are illustrated in figure 2.18. Notice that the shape of the dilated single pixel near the head of the screw is exactly the same as the type of structuring element used to perform the dilation.

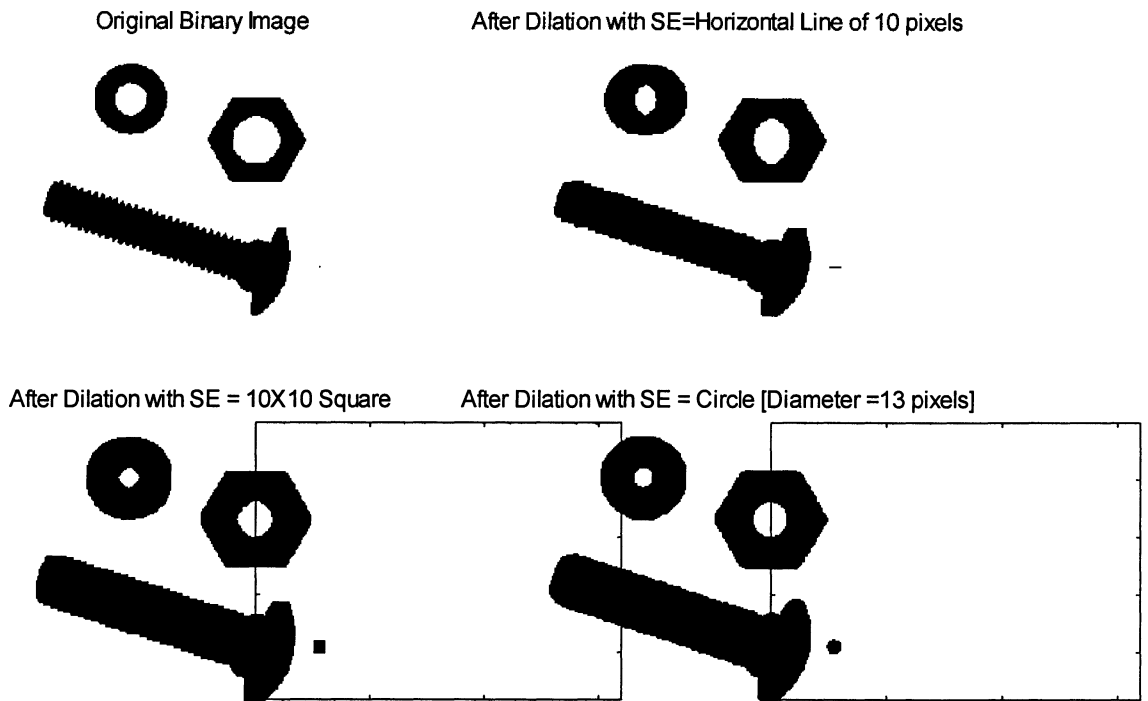


Figure 2.18 Examples of morphological dilation with various structuring elements; (a)

Horizontal line, (b) Square, (c) Disc

Opening and Closing

The morphological operations of opening and closing are considered secondary operations since they are implemented using a combination of erosion and dilation with a particular structuring element. Opening and closing partially reverse some of the changes made by erosion and dilation such that the essential features in the original image are preserved. Like erosion and dilation, opening and closing are complementary operations, in that applying one to the image set A is equivalent to applying the other to the complement set A^c .

An opening is defined as an erosion of an image with a certain structuring element followed by a dilation of the eroded image with the same structuring element. Like erosion, the effect of morphological opening is to remove pixels from objects resulting

from small spurs and noise spikes. However, opening tends to maintain the original shapes of the objects since a dilation operation is performed after the erosion. In set theory notation, if the set S' denotes the reflection of the structuring element set S in the pixel location (i, j) [i.e. the rotation of S through 180° about the reference pixel (i, j)], then the opening of a set A (i.e. a binary image or part of it) by the structuring element set S [i.e. $\psi_S(A)$] is defined by:

$$\psi_S(A) = (A \ominus S) \oplus S'. \quad 2.11$$

The subjective effect of opening using a given structuring element on an image is the same as if we tested whether the structuring element can be moved around the edge of an object in the image in such a way that it is entirely contained within the object. Any pixel hit by the structuring element while it's moving in such a way becomes part of the opening [i.e. set to black (0)]. Figure 2.19 illustrates the opening of the same binary image used in the erosion and dilation operations using the same three shapes of structuring elements. As seen here, the object shapes and sizes are maintained, however their edges have been smoothed to represent the respective shapes of their structuring elements.

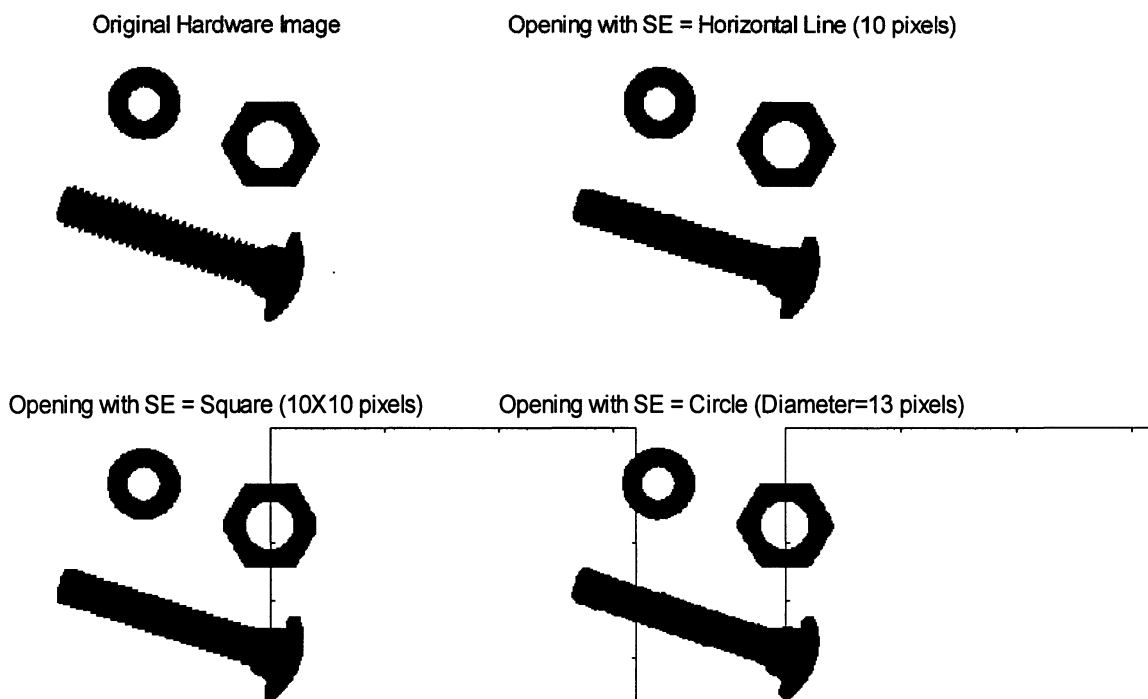


Figure 2.19 Examples of opening operations using various shapes of structuring elements; (a) Horizontal line, (b) Square, (c) Disc

A closing, on the other hand, is defined as a dilation of an image with a particular structuring element, followed by an erosion of the dilated image with the same structuring element. The closing of a set A (i.e. a binary image or part of it) by the structuring element S [i.e. $\phi_S(A)$] is defined by:

$$\phi_S(A) = (A \oplus S) \ominus S. \quad 2.12$$

Subjectively, closing of an image using a given structuring element is the same as if one tested whether the structuring element can be moved around the object edges in such a way that it is entirely contained within the background while touching any part of the object edge. Any pixel hit by the structuring element while it is moving in such a way is a part of the closing (i.e. set to black). As a result, the original shapes and sizes of the objects are preserved, however small holes and gaps between the objects are filled. As

with the opening operations, closing also smoothes object edges to resemble the shapes of the respective structuring elements used for the operation. An example of the closing operation used on the same binary image with the same three types of structuring elements is illustrated in figure 2.20.

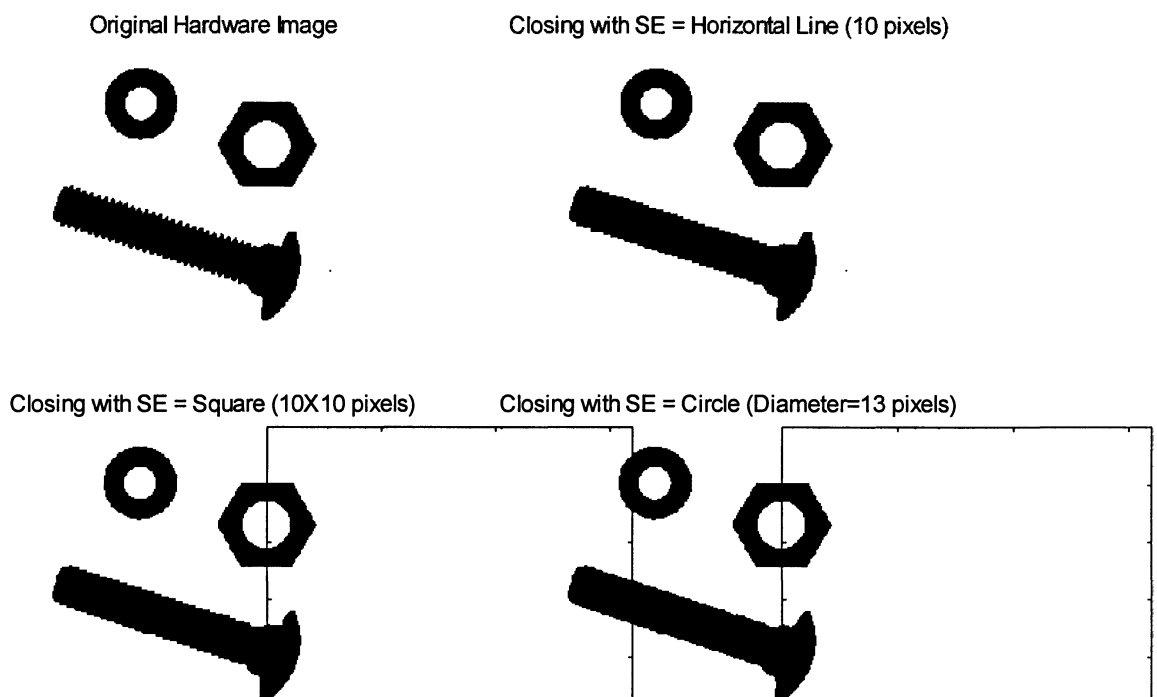


Figure 2.20 Examples of closing operations using various shapes of structuring elements; (a) Horizontal line, (b) Square, (c) Disc

Morphology of Grayscale Images

Morphological operations discussed so far were developed for binary images, however their implementation can be extended to grayscale images as well. Many analogous operations corresponding to binary image morphology exist for grayscale images. Generally grayscale morphological operations are used in images which cannot be primarily converted to binary images due to significant degradations in the image.

This occurs when gray levels vary widely within the object or background portions of an image. Input and corresponding output images are of the grayscale form in grayscale morphology. This means that both input and output pixel values range from 0.0 (black) to 1.0 (white), in the case of an intensity image, or from 1 (black) to 256 (white), in the case of an indexed image. To illustrate the various grayscale morphological operations in this sub-section only intensity images are used. For grayscale morphological operations an object is defined as a group of pixels having brightness values which are greater than the brightness values of the background. Thus objects are brighter than the background in grayscale images. The higher the separation between the object and background brightnesses, the better is the morphological operation.

Like spatial convolution and binary morphology, the grayscale morphological operation moves across the input image pixel by pixel, placing resulting pixels in the output image. At each input pixel location, the input pixel and its neighboring pixels are evaluated using a structuring element to determine the output pixel's brightness value. The structuring elements for grayscale morphology can have various shapes and sizes depending upon the desired results after the operation is performed. An array of coefficients ranging between -1.0 and 1.0 defines the structuring element of grayscale morphology. The structuring element is placed over an input image with its reference coefficient aligned with the input pixel. The reference and its neighboring coefficients are each added to the corresponding input pixel and each of its neighbors. Depending upon how the output pixel value is determined differentiates grayscale erosion and dilation operations.

As in the case of binary morphology, grayscale erosion reduces the size of objects relative to their background, whereas the dilation operation expands the size of the objects. For the erosion case, structuring element coefficient values can range from -1.0 to 0.0, but are generally 0. As a result, the general 3 x 3 square grayscale erosion structuring element looks like:

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

The central coefficient is the reference value in the above case. For grayscale erosion, the reference coefficient and its eight neighbors are each added to the corresponding input pixel and its neighborhood. The output pixel value is the *minimum* of the nine addends. This process repeats pixel by pixel, for each pixel in the input image. As a result, grayscale erosion has the effect of darkening the bright objects in an image, thus making them appear smaller [Baxes, 1994]. The erosion operation has the overall effect of reducing the brightness of the grayscale image. Since all structuring element (morphological mask) coefficients are 0 in the above case, no change would be implemented in the input image brightness when the mask is on top of a constant region. However, in any other region the output pixel value will assume the brightness of the darkest pixel in the input pixel group. Bright objects shrink in size and very small bright objects will completely disappear in the case of grayscale erosion. Figure 2.21 illustrates the grayscale erosion of an 8-bit intensity image with a square structuring element having the above mentioned dimensions and coefficient values.

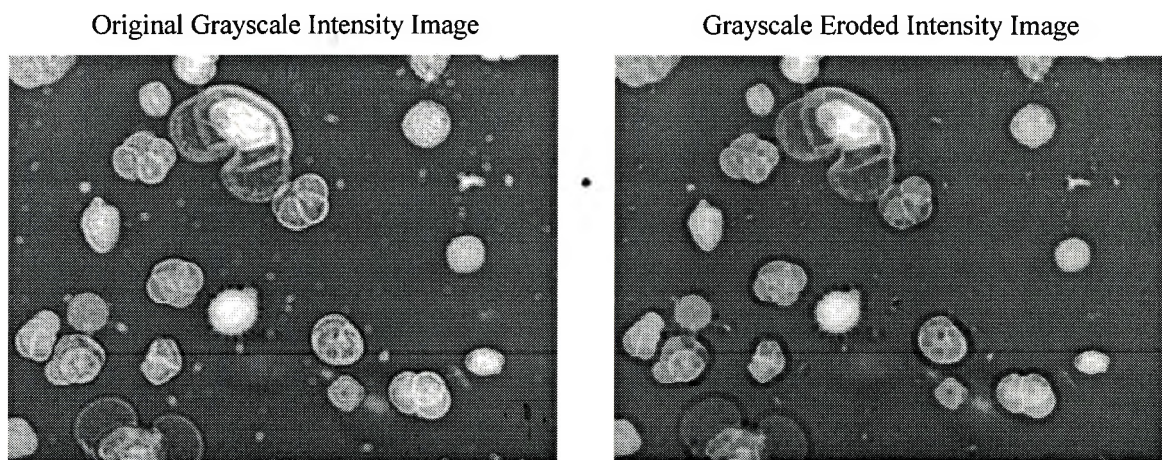


Figure 2.21 Example of grayscale erosion using a square structuring element

By varying the structuring element coefficient values between -1.0 and 0.0, one can achieve different effects. If all the coefficients were decreased to a lower value (e.g. -0.3) a quicker erosion of the grayscale image would result. This is because, before the minimum operator, the negative coefficients are first added to the input image.

In the case of grayscale morphological dilation, the structuring element coefficient values can range from 0.0 to 1.0, but are generally 0. As a result, the general 3 x 3 square grayscale dilation mask is identical to that of grayscale erosion [Baxes, 1994]. However, in this case the output pixel brightness value is determined by the *maximum* of the nine addends of pixels and corresponding coefficients. The subjective effects of grayscale dilation are brightening the bright objects, and thus making them appear larger. The overall image brightness also increases as a result. Like the erosion mask, the output pixel values remain unchanged when the structuring element is applied to regions of constant brightness. However, in any other region the output pixel will assume the brightness value of the brightest pixel in its neighborhood. The sizes of objects are expanded by grayscale dilation, and very small objects become larger with exaggerated shapes. As in the case of grayscale erosion, one can vary the coefficient values in the structuring element between 0.0 and 1.0 to achieve different effects. By increasing all the coefficient values (e.g. 0.3) one can achieve a quicker dilation. An example of grayscale dilation, on the same 8-bit intensity image with the same structuring element as the previous figure, is illustrated in figure 2.22.

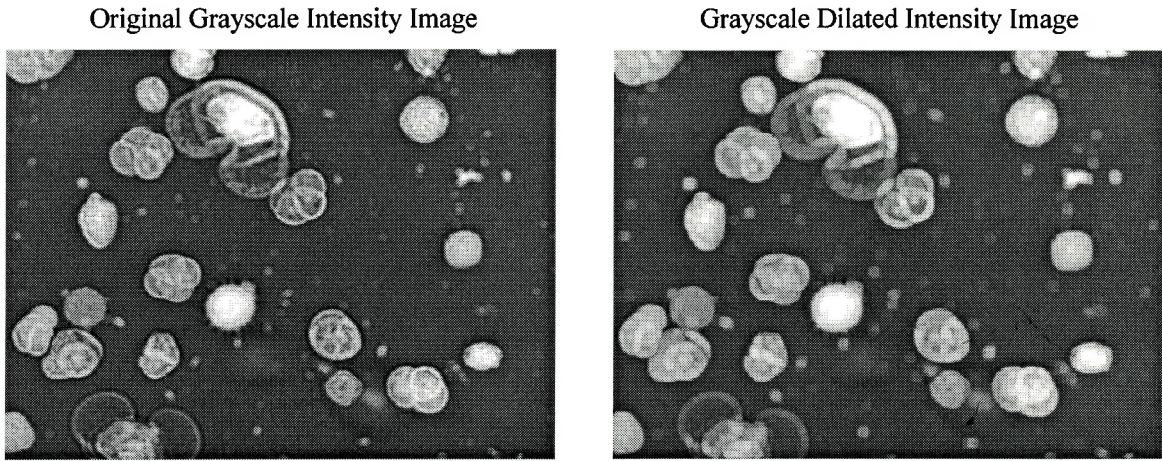


Figure 2.22 Example of grayscale dilation using a square structuring element

In set theory notation, grayscale erosion and dilation may be represented as follows:

$$(f \ominus S)_{ij} = \min_{(k,l) \in S} f_{i+k,j+l}, \quad 2.13$$

$$(f \oplus S)_{ij} = \max_{(k,l) \in S} f_{i+k,j+l}, \quad 2.14$$

where $(k, l) = (0,0)$ is the reference pixel in the structuring element set S ; and f_{ij} is the intensity of the input pixel at spatial location (i, j) . Thus one is transforming f by taking the minimum or maximum of f in a neighborhood, corresponding to the region S , about each pixel of f [Glasbey et. al., 1995].

Grayscale opening and closing are implemented similar to their binary counterparts which have been discussed earlier. A grayscale opening operation is a grayscale erosion with a particular structuring element, followed by a grayscale dilation of the eroded image with the same structuring element. This operation darkens small objects and removes single pixel objects resulting from noise spikes and spurs. However, due to the grayscale dilation as a second step in the opening operation, objects tend to retain their original shapes and sizes. Figure 2.23 illustrates a grayscale opening of the same 8-bit intensity image as the one used to illustrate grayscale erosion and dilation.

Once again a 3 x 3 square structuring element with zero value coefficients is used to perform the operation.

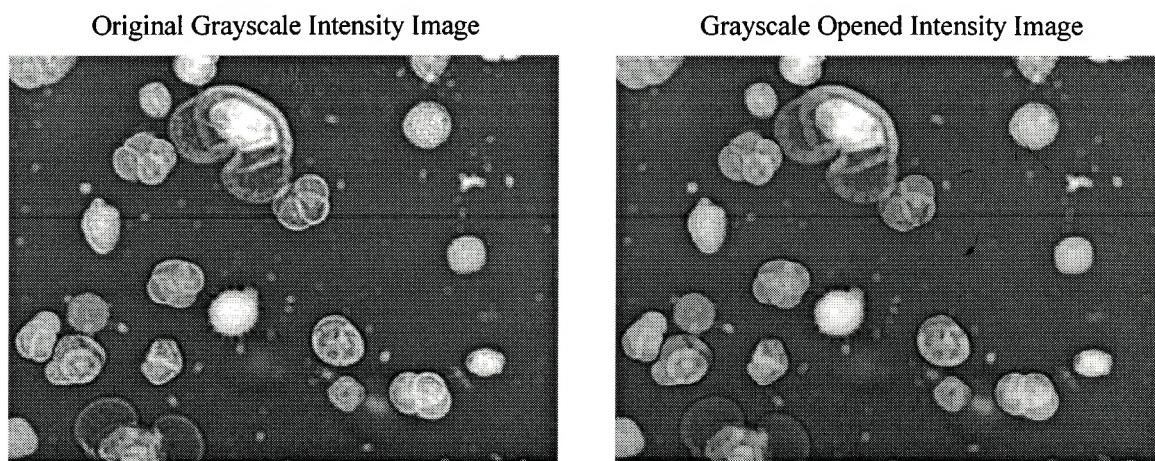


Figure 2.23 Example of grayscale opening using a square structuring element

The grayscale closing operation is a grayscale dilation of an image with a particular structuring element, followed by a grayscale erosion of the dilated image using the same structuring element. This operation is responsible for brightening small objects in a grayscale image and filling-in single pixel objects like small holes and gaps within objects. Like the binary version, objects tend to retain their original sizes and shapes. Figure 2.24 illustrates an example of grayscale closing on the same 8-bit intensity image, and using the same structuring element, as the previous figure.

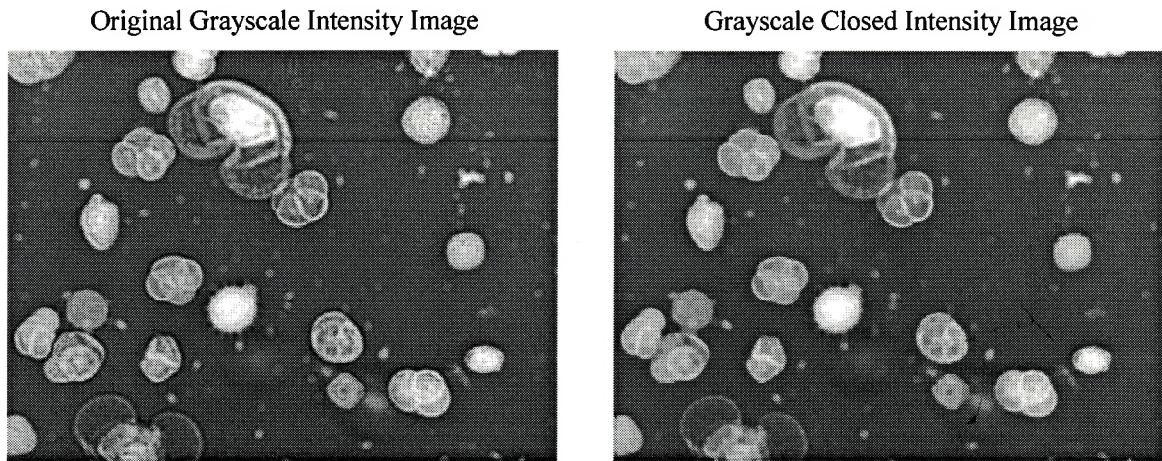


Figure 2.24 Example of grayscale closing using a square structuring element

In set theory notation, the operations of grayscale opening and closing are the same as those described in the case of binary images [i.e. $\psi_S(f) = f \ominus S \oplus S'$ and $\phi_S(f) = f \oplus S \ominus S'$, respectively].

Morphological operations in both binary and grayscale images may be used for the purposes of image enhancements like edge detection, image texture and topology, etc. Several methods like Top-Hat and Well Transforms, Watershed Edge Detection, Distance Transforms, etc., which solely use morphological operations, have been developed over the past 20 years to enhance images. The interested reader may refer to the following references for further details regarding these and more methods of image enhancement using morphological operations [Glasbey et. al., 1995; Baxes, 1994; Serra, 1982; Berman et. al., 1994].

2.3.3 Object Classification and Measurement

After the processes of image segmentation and morphology, which are used to cleanly divide an image into discrete objects, the next step is the extraction of quantitative information from the images by measuring individual features of the objects. This information can be compared to known measures of objects to classify them into one of

several categories. Object measurement and classification is regarded as the endpoint of much image analysis. Most of the object measurement techniques are concerned with measuring various attributes of objects contained within the bounds of an outline.

By measuring the brightnesses of each pixel contained within the object's boundaries one can develop several statistics to measure the brightness features of an object. The mean, mode, and standard deviation of the brightnesses within objects are useful measures of the brightness and color features of objects. The surface finish of a smooth or coarsely textured object within an image may be measured by examining its spatial frequency content. Smooth objects display small variation in their brightnesses thus exhibiting low spatial frequencies, whereas coarsely textured objects exhibit high spatial frequencies due to a high brightness variance. One may use low-pass or high-pass filters to attenuate the respective spatial frequencies to determine the object's texture. Alternatively, Fourier transforms may be used to perform the same task.

The most common object measurements made are those pertaining to the object's size and shape. The two most widely used statistics to describe an object's size are measurements of area and distance. These measurements can be further divided into various sub-measures which can collectively be used to indicate both size and shape information of objects. However, in any image analysis application the goal is to use the fewest necessary measures to characterize an object adequately so that it may be unambiguously classified as required by the application [Baxes, 1994]. Some of the most common object size and shape measures are briefly discussed below.

The object *area* is simply the pixel area of the interior of the object. It is computed by summing the total number of pixels included in the object boundary. This is an ideal measure of the object size. However, the calculated area does not include the areas of holes within the object.

The *perimeter* of an object is the pixel distance around the circumference of the object. It has been theoretically proven [Glasbey et. al., 1995] that accurate computation of this measure is determined when an object boundary pixel contacts its neighbor

vertically or horizontally, the pixel distance be set to 1 unit. However, when an object boundary pixel contacts a neighbor diagonally, the pixel distance be set to $2^{0.5}$, or 1.414 units. This results in a measure of the boundary length of the object.

One of the most commonly used shape statistics is a measure of *compactness*. This is defined to be the ratio based on the area and the perimeter measures of an object. Compactness can also be thought of as a ratio of an object's area to the area of a circle with the same perimeter. A circle is used as the basic shape with which to make the comparison since it is the object with the most compact shape [Glasbey et. al., 1995]. The ratio varies between 0 and 1. The greater the ratio, the rounder the object. The statistic is given by the following equation:

$$\text{Compactness} = (4\pi \times \text{Area}) / \text{Perimeter}^2 \quad 2.15$$

The *major axis* of an object is defined by the (x, y) endpoints of the longest line that can be drawn through the object. These endpoints are found by computing the largest pixel distance between every combination of border pixels in the object. The length of the major axis is usually determined by the euclidean distance between its endpoints:

$$\text{Major Axis Length} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad 2.16$$

where (x_1, y_1) and (x_2, y_2) are the major axis endpoints. This measure gives an idea of the length of the object. The angle between the major axis and the x-axis (horizontal dimension) of the image is the *major axis angle*. This angle ranges between 0° and 360° , and gives a measure of the object's orientation. The major axis angle of an object is given by:

$$\text{Major Axis Angle} = \tan^{-1} \left(\frac{(y_2 - y_1)}{(x_2 - x_1)} \right) \quad 2.17$$

The *minor axis* of an object is determined by finding endpoints of the longest line which can be drawn through the object while maintaining perpendicularity with the major axis. The euclidean distance is also used to measure the length of the minor axis. A good measure of the elongation of the object is the ratio of the minor and major axes lengths.

A ratio of approximately 1 would mean that the object is roughly circular or square. Elongation becomes pronounced as this ratio decreases from 1. The area of a box that would entirely surround the object is called the *bounding box* area. Its dimensions are those of the major and minor axes.

A measure of the area of holes within an object is necessary to determine the size of the object's hole content. The Euler number of an image is defined to be the number of objects minus the number of holes within an image. A ratio of the total hole area to object area gives a measure of the perforation of the object. A ratio of 1 would mean that the object is entirely a hole.

Spatial moments of an object are statistical shape measures that do not specifically characterize the object, rather they give statistical measures which are related to the object's characterization [Baxes, 1994]. The sum of the pixel brightness values in an object is defined as the zero-order spatial moment. This measure is related to the energy, or aggregate brightness of the object. The first-order spatial moments of an object are the separate x and y sums of pixel brightnesses in the object, each multiplied by its respective x and y coordinate location in the image. The first-order spatial moments of an object represents the object's mass distribution through its area. The center of mass (*centroid*) of an object, which is the balance point of the object, is determined by using the first-order spatial moments.

Centroid x -coordinate = Sum of object's x -pixel coordinates / No. of pixels in object

Centroid y -coordinate = Sum of object's y -pixel coordinates / No. of pixels in object

Boundary descriptions of the object are used to define the outline of an object and record specifically how this outline behaves. There are several boundary descriptors used to measure the outlines of an object within an image. Out of these measures *relative chain codes* is the most computationally efficient method to record the movements of the object outline. This method records the relative direction from the chosen reference

boundary pixel to the next, depending upon which of the eight adjoining pixels is the boundary of the object continued. For every move in a particular direction from the reference pixel a direction code (1 to 8) is recorded which indicates the direction followed. However, this method is relative to the direction followed by the previous step. As a result, once the codes have been recorded, there is no information about the orientation of the boundary of the object. This feature is extremely important in object classification with the help of machine vision. Further details regarding this and other measures of the object boundary e.g. line segment representation, Fourier descriptors, etc. may be sought from [Baxes, 1994; Glasbey et. al., 1995; Gonzalez et. al., 1987].

After segmenting the image into discrete objects, measuring the objects using a variety of shape measurements including specific size and shape dimensions, shape statistics, and object boundary descriptions, one has to classify these measurements. This is the final step in the traditional image analysis process. Object classification is used primarily in the field of machine vision and image interpretation. This process involves the comparison of a set of measured features of an object with some established criteria. Depending upon the tolerance limits set by a particular application, a comparison is made to determine whether the measured features lie within the required specifications. Generally, the measures which are computationally the easiest to derive are used to classify an object. Measurement invariance of an object is of the utmost importance while classifying it. This is because in machine vision variations in the object's orientation, scaling, spatial location etc. should not affect the classification process. As a result, such measures which are invariant to these common variations in the object (e.g. relative chain code, relative line segment, normalized object dimensions etc.) should be used for classification. For further details regarding traditional object classification techniques one may refer to any basic text in the image processing literature. Advanced object classification techniques using fuzzy logic and neural network concepts have recently been evolving. These methods use more advanced techniques to determine a match between an object and a class. For example, neural networks can be "trained" by

showing them several images of known object characteristics and specifying their correct classifications. In this way the network learns the feature measures and the tolerances for each classification category [Baxes, 1994]. Image classification using multivariate statistical methods will be a topic of future research in this project. These statistical methods are useful when there are a large number of digital images which need to be classified into various categories depending upon the specifications of their respective applications.

Before concluding this chapter, it should be noted that traditional image processing has two more fundamental classes besides the three discussed in this chapter. These two classes are image compression, and image synthesis. Since the operations involved in these classes of traditional image processing are not directly concerned with the goals of this research, they have been excluded. The reference section at the end of this thesis provides a good list of references which extensively cover all five classes of traditional image processing. This list can be used as a starting point by any reader who knows very little regarding the realm of traditional image processing techniques.

3. MULTIVARIATE IMAGE ANALYSIS (MIA) USING STATISTICAL LATENT VARIABLE METHODS

Digital image analysis techniques are finding considerable use in a growing number of industrial processes. On-line imaging sensors are beginning to replace sensory equipment to monitor various manufacturing processes. Examples include cameras used to monitor the state of combustion in furnaces, or to monitor sizes and shapes of machine-made parts, and laser induced scanners to monitor surface properties and other faults in sheet-forming processes. Since most industrial processes are time-varying in nature, this chapter addresses the use of digital image analysis techniques for on-line fault detection and monitoring of time-varying processes.

Since imaging sensors collect data in the form of digital images, industrial processes equipped with such sensory equipment are collecting vast amounts of data. Due to this massive data overload, along with the need to get fast results, the most commonly used image analysis techniques are very simple. Other than a basic visual inspection of the digital image by the operators, the most commonly used image analysis techniques for fault detection are restricted to thresholding and area counting. These simple analyses, along with some basic statistical calculations on the acquired image, are deemed adequate as far as the needs of many industrial processes are concerned. Unfortunately, these results come at the expense of a huge information loss, which could sometimes be vital in investigating the overall performance of the processes. There is a lot of information in the visual data which could potentially be used to accurately determine if the process is functioning fault-free and efficiently. Recently, multispectral imaging sensors (which have the ability to acquire visual data at multiple wavelengths)

have been developed. As a result, the amount of input data and the information content, have increased even more.

It would be a great asset if the full information content of such data could be explored quickly and easily. Vital information regarding process performance includes prompt and accurate identification of the nature of faults as well as their exact locations while the process is on-line. This chapter investigates the use of a relatively new and powerful statistically driven digital image analysis technique called Multivariate Image Analysis (MIA). The purpose of MIA is to extract useful information from visual data transmitted by multispectral imaging sensors. This information can then be interactively used in both monitoring and feedback control schemes.

The first part of this chapter provides an overview of Multivariate Image Analysis (MIA) with a brief literature review of both the theory and applications. Some of the ideas introduced in the overview are illustrated through an application of MIA in the area of remote sensing. The theory and ideas of MIA are extended to form the basis for an on-line monitoring scheme in the second part of the chapter. This scheme is successfully applied to monitor and locate specific landscape features on a LANDSAT (MSS) satellite image example. Finally, from the successful applications of MIA, conclusions are drawn on possible extensions of MIA for fault detection, on-line monitoring, and spatial feedback control schemes in industrial processes.

3.1 Overview of Multivariate Image Analysis

Image data when collected in multiple spectral bands produces a multivariate image. Different scientific fields have adapted the concepts of multivariate imaging after the systematic introduction of multiple wavelength images in the late 1970's by the satellite imaging community. A concise history of multivariate imaging is provided by Geladi et al. [1996]. They also discuss the various methods and types of multivariate imaging which is used in fields ranging from remote sensing to electron microscopy.

A multivariate image consists of a stack of congruent images, where each image in the stack is measured for a different wavelength, frequency, or energy [Geladi et al., 1996]. Figure 3.1 illustrates a stack of 4 congruent 512 x 512 pixel images, with each image having a unique wavelength.

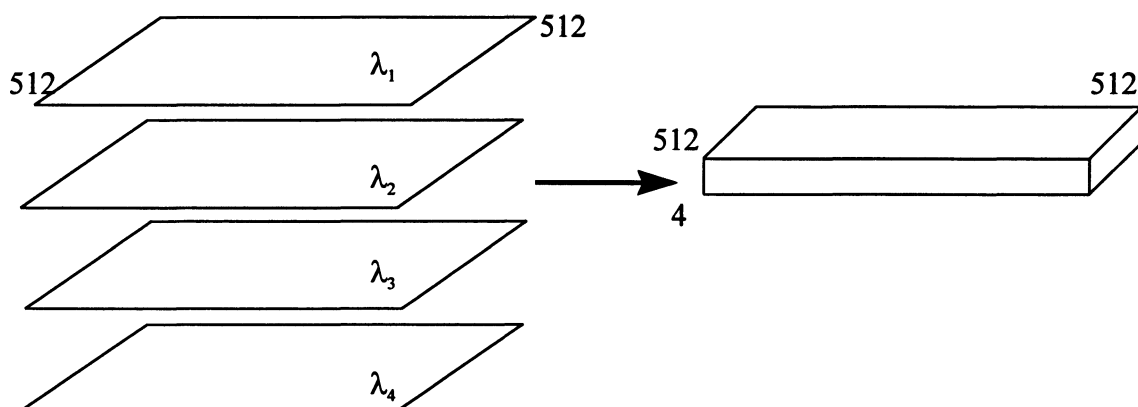


Figure 3.1 A stack of 4 congruent images (collected in 4 different wavelengths) to form a 512 x 512 x 4 Multivariate Image

A multivariate image can be treated as a three-way data array of pixel intensity values where a third pixel coordinate (z) is added to the usual coordinates (x and y). Thus, (z) represents an index for 'variables' in the multivariate image. Alternately, one could view a multivariate image as a two-way array of variable vectors with each vector representing

a pixel [Geladi et al., 1989]. Figure 3.2 illustrates the same $512 \times 512 \times 4$ multivariate image from the previous figure as a two-way array of variable vectors at their respective pixel locations.

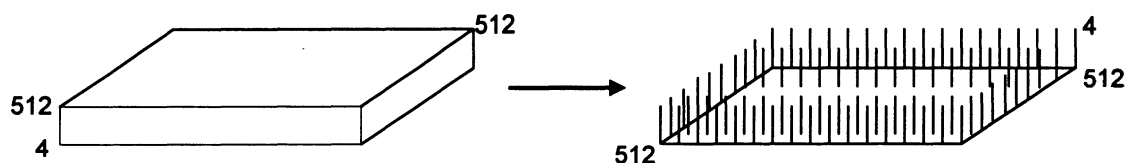


Figure 3.2 A $512 \times 512 \times 4$ Multivariate Image viewed as a two-way array of 4×1 variable vectors in the image plane. For graphical clarity not all variable vectors are shown.

Congruence in imaging is defined by Geladi et al. [1996] as two or more stacked images such that for each pixel in one image there is a corresponding pixel in the other images that can be referred to the same position in the object or scene depicted. Since data in multivariate images consists of several congruent images, each variable vector in such an image contains highly correlated pixel intensity values. Furthermore, a common multivariate image usually contains an enormous amount of data (e.g. $512 \times 512 \times 4 = 1,048,576$ pixel intensities), making the analysis of such images computationally intensive if one were to use traditional image processing techniques. As a result, there is a definite need for some data analysis techniques that can handle large volumes of highly correlated data. Several data analysis techniques exist that can handle multi-way data arrays. However, certain statistical methods are more ideally suited to analyze multivariate images due to the unique characteristics of the data forming these images.

Latent variable statistical methods like Multi-Way Principal Component Analysis (PCA), Partial Least Squares or Projection to Latent Structures (PLS) have been successfully applied to analyze multivariate image data. These methods efficiently compress highly correlated data and produce it onto a reduced dimensional sub-space through a few linear combinations of the original data. The main objective of these

methods is to explain the variance-covariance structure of the multivariate data. With highly correlated variables, one usually finds that only a few reduced dimensions (principal components) are needed to explain most of the significant variation in the data [Nomikos, 1995]. A vast amount of literature has been dedicated to both the theory and applications of latent variable methods. Further details on these methods may be obtained from various sources, some of which are [Jackson, 1991; Jolliffe, 1986; Wold et al., 1987; Hoskuldsson, 1988]. In this work the main focus shall be the application of latent variable methods to multivariate images. It is assumed that the reader has some basic background in multivariate statistics (particularly PCA) while reading the techniques described in this chapter.

As explained in the traditional image processing section of this report, a digitized image can be treated as a two-way array of pixel intensities. This image can be viewed as a matrix \mathbf{X} having two indices n_x and n_y . When many variables are involved a third index n_z is added to form a three-way array which is represented as $\underline{\mathbf{X}}$. Traditional latent variable methods decompose data from two-way arrays with M objects (rows) and N variables (columns) into a latent structure (given as an outer product of score vectors \mathbf{t}_a and loading vectors \mathbf{p}_a) plus a noise part \mathbf{E} [Geladi et al., 1989]:

$$\mathbf{X} = \sum_{a=1}^A \mathbf{t}_a \mathbf{p}'_a + \mathbf{E} = \mathbf{TP}' + \mathbf{E} \quad 3.1$$

where A is equal to or less than $\text{rank}(\mathbf{X})$; \mathbf{X} = two-way data matrix ($M \times N$); \mathbf{E} = residual matrix ($M \times N$); \mathbf{T} = score matrix ($M \times A$), where the columns \mathbf{t}_a are the score vectors for component a ($M \times 1$); \mathbf{P}' = loading matrix ($A \times N$), where the rows \mathbf{p}'_a are the loading vectors for component a ($1 \times N$). The score vectors are linear combinations of the variables (columns) in the data matrix \mathbf{X} that explain the greatest variation in the multivariate data. These vectors have a property of orthogonality with respect to each other. Loading vectors are the eigenvectors (in descending order) of the variance-covariance structure ($\mathbf{X}'\mathbf{X}$) in the data matrix. These vectors have the property of orthonormality with respect to each other (i.e. $\mathbf{P}'\mathbf{P} = \mathbf{I}$; where \mathbf{I} is the identity matrix)

[Geladi et al., 1996]. Using the above equation, along with the properties of the score and loading vectors, it is easily shown that scores can be calculated from loadings and vice versa:

$$\mathbf{T} = \mathbf{X}\mathbf{P} \quad 3.2$$

In order to use latent variable methods on multivariate images one needs to reorganize the three-way data array into a corresponding two-way array containing a 'variable' (column) part and an 'object' (row) part. Three-way arrays can be reorganized into corresponding two-way arrays in three different orientations. However, two of these orientations reorganize the data such that the 'variable' way in the multivariate image (n_z) corresponds to the columns in the reorganized two-way data array. The objects (rows) in this reorganized array correspond to ($n_z \times 1$)' vectors from figure 3.2 throughout each pixel location from the image plane in the multivariate image. As a result, the three-way array $\underline{\mathbf{X}}$ having dimensions n_x , n_y , and n_z is reorganized into a two-way array \mathbf{X} having dimensions ($n_x \cdot n_y$) and n_z . In most multivariate images, ($n_x \cdot n_y$) is extremely large and n_z is relatively small (long and thin reorganized data arrays). Upon completion of the reorganization step, traditional latent variable methods can be used to decompose the resulting array into a reduced dimensional sub-space plus a noise part. The reorganization and decomposition (using PCA) of a three-way array having dimensions of 512 x 512 x 4 is illustrated in figure 3.3. Reorganization of the three-way array into a matrix (two-way array) is a great simplification. As a result, some of the difficult three-way algebra, while using latent variable methods to decompose the multivariate image, is avoided. This reorganization can give a loss of image context; however, this loss is only temporary since the score vectors can themselves be further reorganized into images.

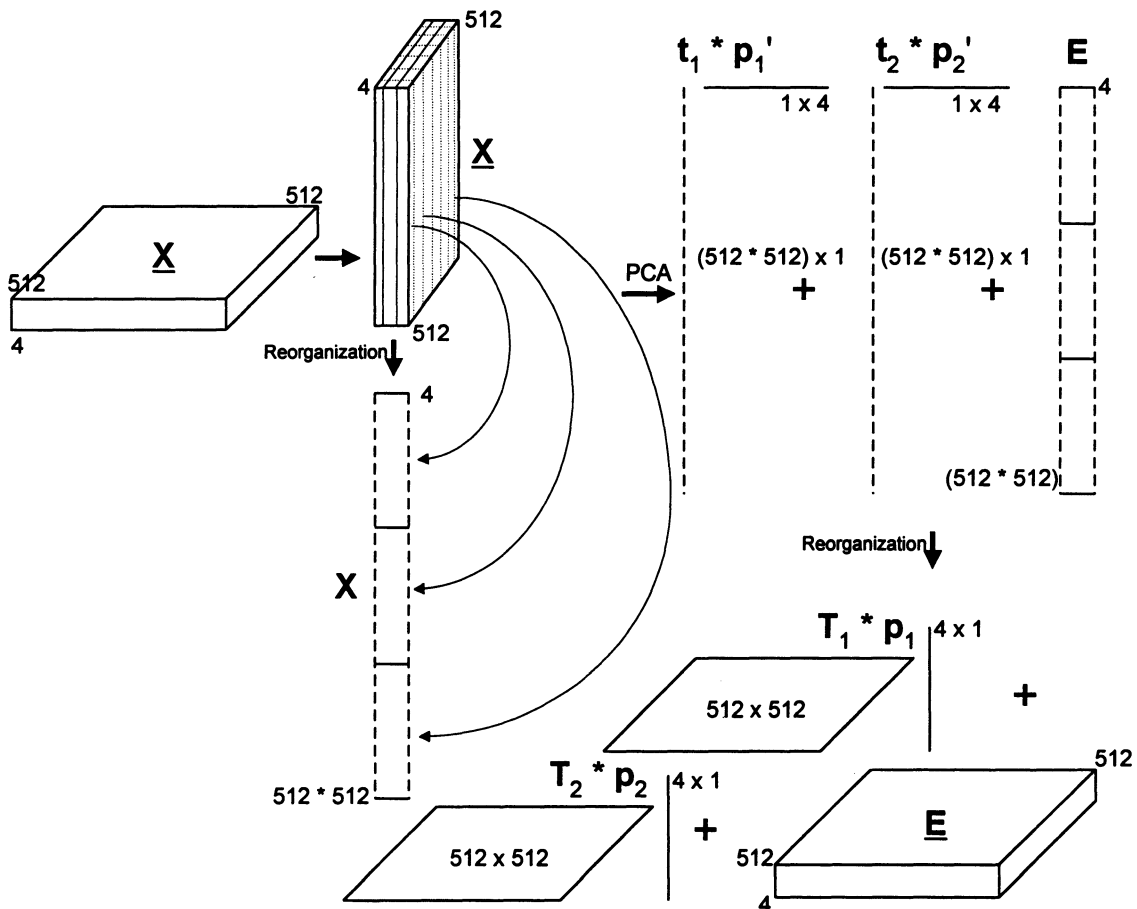


Figure 3.3 A 512 x 512 x 4 multivariate image reorganized into a $(512 \cdot 512) \times 4$ array followed by PCA decomposition into reduced dimensional sub-space

In addition to compression via reducing the dimensionality of the highly correlated data, latent variable statistical methods also serve as efficient information extractors from multivariate images. Besides the obvious information content which is distributed throughout the multivariate image, individual variable images contain unique information which is specific only to that particular wavelength, frequency, energy etc. Since latent variable methods extract linear combinations of the highly correlated variable data in a multivariate image, the full information content is efficiently pooled together to form the latent variables. Each latent variable extracts a particular spectral feature (a linear

combination of intensities over all wavelengths) from the image, and the score plot for that dimension is a representation of the image in terms of that spectral feature.

Mean-centering and scaling of the two-way data array before performing PCA or PLS is prescribed in the classical latent variable methods literature. However, this is difficult (and irrelevant) in the context of MIA due to the nature of the numbers (pixel intensities, e.g. 0-255) in the array. As a result, to avoid numerical difficulties, and to minimize computing time, mean-centering and scaling are usually excluded in MIA. Geladi et al. [1996] do give some examples of simple and non-linear scalings that can be applied to multivariate images in special cases. In this report mean-centering and scaling of multivariate images prior to using latent variable methods is omitted.

As explained earlier, it is usually necessary to reorganize the three-way array $\underline{\mathbf{X}}$ into a corresponding two-way array \mathbf{X} prior to performing PCA or PLS on the multivariate image. Consider an example multivariate image with variable images of dimension 512 x 512 pixels in Q different wavelengths. It can be seen from figure 3.3 that the resulting reorganized two-way array would be a matrix with Q variables (columns) and 262,144 objects (rows). The direct use of latent variable methods on such large data arrays can pose some computational difficulties. These problems can be alleviated by either using representative samples from the data array (i.e. sub-sampling), or carrying out calculations on the $\mathbf{X}'\mathbf{X}$ matrix [Geladi et al., 1989]. The $\mathbf{X}'\mathbf{X}$ matrix is called the cross-product matrix. This matrix has dimensions of $Q \times Q$ (no matter how large the number of objects in the rearranged two-way array) and is considerably smaller than \mathbf{X} . Despite its small size, the cross-product matrix still contains all the necessary information to calculate the loading vectors of \mathbf{X} . A numerical decomposition technique called Singular Value Decomposition (SVD), which is similar to PCA, is used to extract the loading matrices \mathbf{P} and \mathbf{P}' from the cross-product matrix:

$$\mathbf{X}'\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{P}' \quad 3.3$$

where \mathbf{P} and \mathbf{P}' = loading matrices of size $(Q \times Q)$ containing the loading vectors of \mathbf{X} as columns and rows, respectively; \mathbf{D} = diagonal matrix of size $(Q \times Q)$ containing the

eigenvalues of $X'X$ on its diagonal in descending size order [Geladi et al., 1989]. Once the loading matrix has been calculated, the score matrix T can be obtained by multiplying P to the reorganized two-way array X (equation 3.2). SVD decomposes a two-way data array into all of its eigenvectors and eigenvalues placing them in descending eigenvalue size order. The sum of all the eigenvalues in the diagonal matrix D is the total sum of squares of the multivariate image. To determine the relevant number of eigenvalues that explain most of the variation in the multivariate image data, one needs to look at the percentage of the total sum of squares carried by each of the eigenvalues in D . Usually the first few eigenvalues account for a large percentage of the total sum of squares. The corresponding eigenvectors of these eigenvalues are regarded to be the relevant loading vectors of the reorganized two-way array X . The remaining eigenvectors are attributed to the noise part of the latent variable decomposition. Using only the relevant number of loading vectors to form the loading matrix P , one can calculate the corresponding score matrix T which contains only the relevant score vectors. The combination of these two matrices forms the reduced dimensional sub-space of the multivariate image (figure 3.3). The only computationally intensive step in the above procedure is the initial calculation of the cross-product matrix $X'X$ from the reorganized two-way array X . Further details on the calculation of the loading and score matrices from the reorganized two-way array using SVD on the cross-product matrix are provided in Geladi et al. [1996].

The above procedure involves pure data analysis techniques, along with matrix algebra, to calculate the reduced dimensional sub-space of a two-way data array. As a result, it can easily be implemented to achieve quick results. However, this procedure gives the impression of a loss of image context since the multivariate image gets decomposed into a combination of score and loading vectors plus a noise matrix. As explained earlier, this loss is only temporary in order to make the decomposition step less computationally intensive and to avoid using complicated three-way algebra.

The resulting score vectors t_a (columns of T) from a latent variable decomposition of a multivariate image have the same *length* $\{(n_x \cdot n_y) \times 1\}$ as the reorganized two-way

array \mathbf{X} (figure 3.3). Furthermore, these score vectors can be folded back up to form two-way arrays (\mathbf{T}_a) having the same dimensions ($n_x \times n_y$) as the variable images in the original multivariate image. These two-way arrays are termed '*score images*', and can be displayed and studied as images. Figure 3.3 illustrates the reorganization of the two score vectors from PCA of a $512 \times 512 \times 4$ multivariate image into a product sum of two score images with their corresponding loading vectors plus a noise part. Some scaling operations are necessary before displaying or storing score images. Since latent variable methods are simply data analysis techniques, the calculated values of the resulting score vectors may not be integer values (as the raw image data). Furthermore, there is no restriction on these methods to avoid negative values. However, image intensities are always positive or zero. Lastly, there might be a size problem if the range of the resulting score vector is too large to fit in the image display. These problems are overcome by appropriate scaling and/or shifting of the data to fit the range of the image display [Geladi et al., 1996].

A lot of interesting details about the information content from the multivariate image under study may be gained by a simple visual inspection of the score images. In this case the values from the score vectors are studied as pixel intensities. Alternately, one can pool and visualize this condensed information from various score vectors by using false-color composites of groups of three selected score images as the red, green, and blue intensity arrays of an RGB image. Since latent variable methods concentrate important information in a few score images, studying these images may prove to be better than simply observing the raw variable images [Geladi et al., 1996]. Furthermore, score images also serve as effective noise and data reducers (if many variables exist) by compressing the multivariate image information content into a few primary latent variable images, while attributing the secondary (higher) latent variables to noise.

To gain further insight into the application of latent variable methods on multivariate images, an example from the remote sensing field is provided. Principal Component Analysis (PCA) has been used to decompose the multivariate image into its

latent variable space. This particular example was chosen due to its excellent variety of potential features (information) that can be extracted into a reduced dimensional subspace. The image used in this example is a LANDSAT (MSS) satellite image of size 512 x 512 consisting of 4 wavelength bands ranging from 500 - 1100 nm. The image has been geometrically corrected with each pixel representing a surface area of 80 x 80 m. This particular multivariate image was acquired from Professor Paul Geladi of Umea University, Sweden and has been used as an example to demonstrate the use of MIA [Geladi et al., 1991; Geladi et al., 1996]. Complete technical details regarding the raw multivariate image data are provided in the above mentioned references. The satellite image depicts a scene of the city of Mobile in Alabama, USA, (in the center of the image) along with the Alabama river delta (towards the top-right corner of the image) and the Gulf of Mexico (towards the bottom-right corner of the image). Figure 3.4 illustrates all four raw variable images. It can be observed that some of the features (e.g. roadways to the city) from the satellite image are specific to particular wavelength bands (variables).

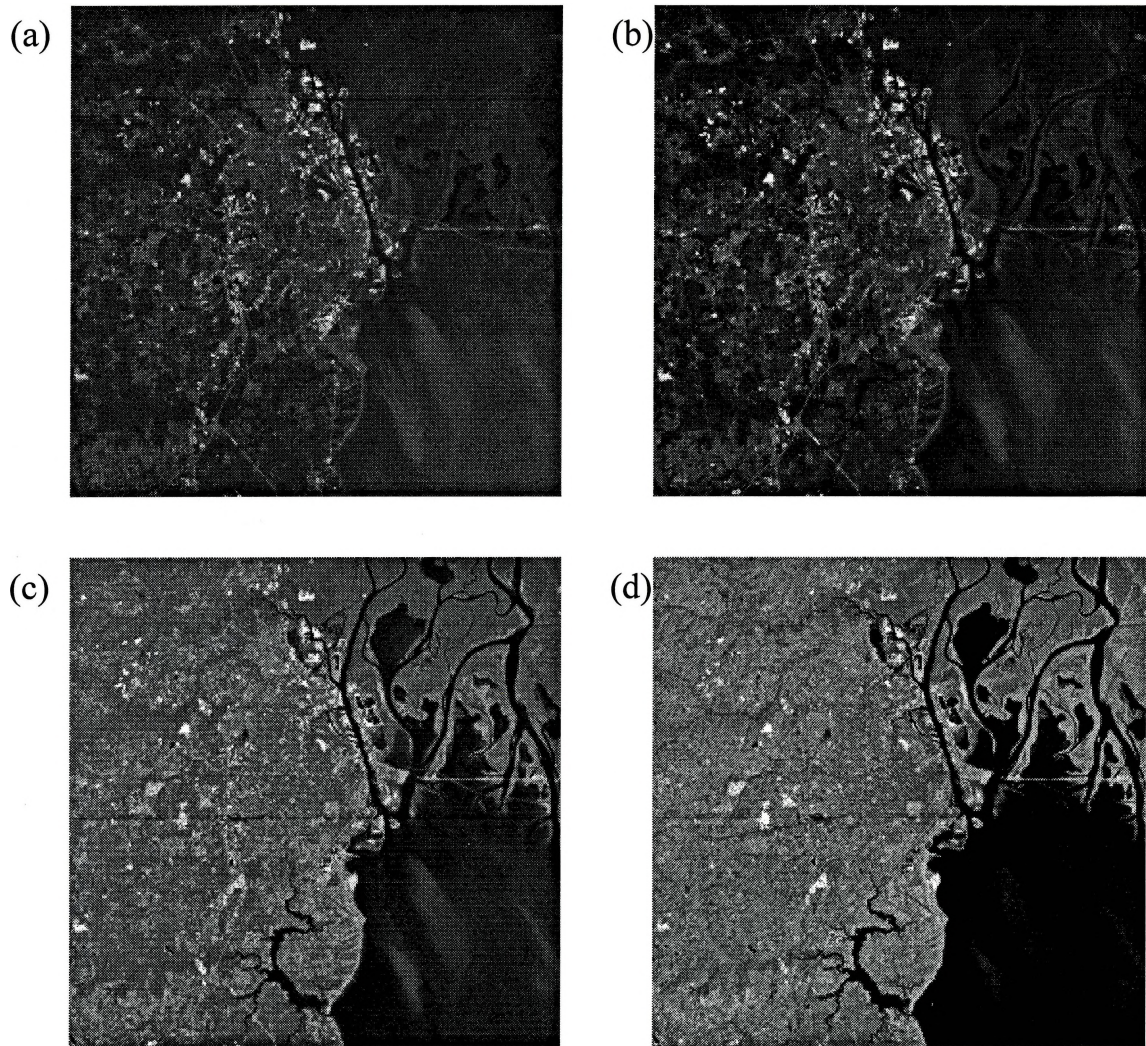


Figure 3.4 A $512 \times 512 \times 4$ LANDSAT (MSS) satellite image. The 4 wavelength bands (variables) are: (a) 500-600 nm {green-yellow}, (b) 600-700 nm {orange-red}, (c) 700-800 nm {deep red}, (d) 800-1100 nm {near-infrared}

The above multivariate satellite image is treated as a three-way array $\underline{\mathbf{X}}$ of size $512 \times 512 \times 4$ with the 4 wavelength bands as the 'variable' way. This array is reorganized into a two-way array orientation \mathbf{X} of size $262,144 \times 4$. The cross-product matrix $\mathbf{X}'\mathbf{X}$ of this reorganized array is only of size 4×4 . This matrix is calculated without any mean-centering or scaling of \mathbf{X} . PCA is carried out on the cross-product matrix using the SVD technique with the help of a high level computer language

{MATLAB™ v4.2c1} to calculate the diagonal eigenvalues matrix **D** and loading matrix **P** of the reorganized two-way array **X**:

$$\mathbf{D} = 1.0 * 10^8 \bullet \begin{bmatrix} 3.807 & 0 & 0 & 0 \\ 0 & 0.191 & 0 & 0 \\ 0 & 0 & 0.008 & 0 \\ 0 & 0 & 0 & 0.004 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 0.356 & -0.550 & 0.743 & -0.137 \\ 0.402 & -0.583 & -0.540 & 0.455 \\ 0.607 & 0.149 & -0.313 & -0.716 \\ 0.586 & 0.579 & 0.242 & 0.512 \end{bmatrix}$$

The score matrix **T** is calculated from **P** and **X** by using equation 3.2. This matrix has dimensions of 262,144 x 4 consisting of four score vectors \mathbf{t}_a as its columns. Individual score vectors are then rearranged to form 4 two-way arrays of size 512 x 512 each. These arrays are scaled appropriately to fit the display range (i.e. integers from 0-255, or real numbers from 0.0-1.0), thus converting them into score images \mathbf{T}_a . Geladi et al. [1989] provide several scaling methods for rearranged score vectors to convert the score values of these matrices into pixel intensities which cover the full display range. Figure 3.5 illustrates the resulting four (scaled) score images for the satellite multivariate image. The space defined by the score images of a multivariate image has been referred as ‘*scene space*’ in some MIA literature [Esbensen et al., 1989] and this terminology will also be used henceforth in this chapter.

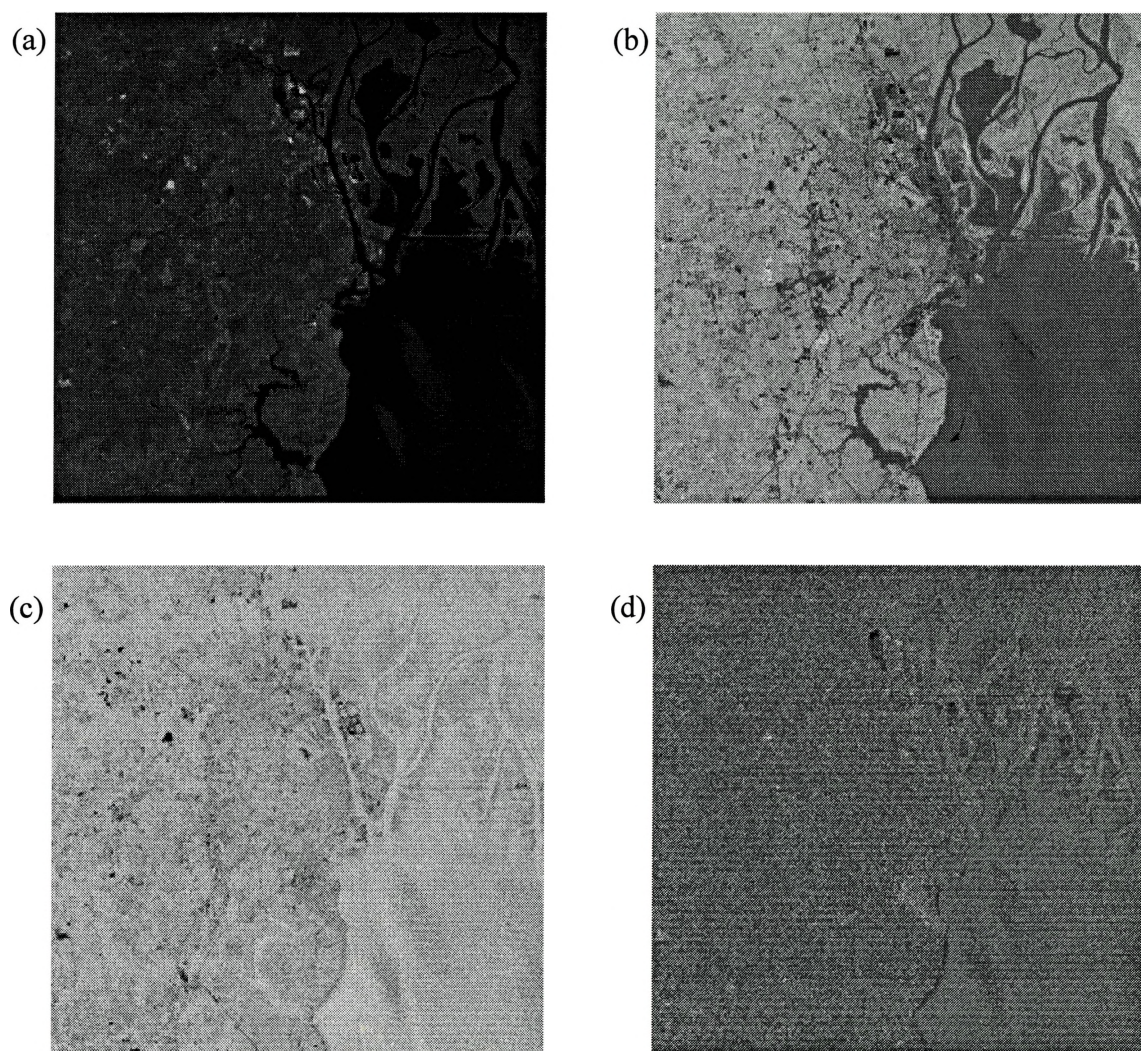


Figure 3.5 The four score images of the $512 \times 512 \times 4$ satellite multivariate image: (a) T_1 , (b) T_2 , (c) T_3 , (d) T_4

Since the raw image data was not mean-centered or scaled prior to performing PCA the first score image is a weighted average of the most intense raw images (see first column of P). Furthermore, this image is the least noisy of the four score images. Thus it can be considered as a noise-reduced mean ('albedo') image. Due to the orthogonality requirements of the score vectors the second score image is in total contrast with the first

image. The third and fourth score images are quite noisy and very little meaningful structural information can be gathered from them [Geladi et al., 1996].

From the diagonal eigenvalue matrix **D** the calculated sum of squares explained by each of the four components of the PCA is 95.0%, 4.7%, 0.2%, and 0.1% from component 1 to 4. Since the original multivariate image has 4 variables, the maximum possible rank of **X** is 4; however, the first two components explain approximately 99.7% of the total sum of squares in the image. As a result, it can be said that the effective rank of **X** is 2, and the remaining 0.3% of the total sum of squares (explained by components 3 & 4) is attributed to noise. This is also apparent from observing the score images in figure 3.5. Thus, components 3 and 4 have been effectively discarded from further analyses.

The scores are the most useful results of the latent variable calculation on multivariate images since they can be used in more than one way. As explained above, scores can be viewed as pixel intensities by converting the score matrices into gray-level or false color composite images. They prove to be effective data reducers, and at the same time, retain the information content from the multivariate image. However, simply observing the score images (or their false color composites) sometimes gives a very subjective interpretation of the useful information in a multivariate image. A more objective analysis of the scene space may be gathered by making scatter plots of one unfolded score vector against another. These plots are usually called score plots in the multivariate statistics community. For A latent variables, the total number of score plots that can be drawn are $A \cdot (A - 1) / 2$ [Geladi et al., 1989]. In terms of MIA the latent variables themselves are images containing a large number of pixels. As a result, a score plot of two 512 x 512 score images would draw 262,144 distinct points. The score values represent linear combinations of the variables (columns representing the 'variable-way') of the reorganized multivariate image **X** throughout its pixel locations (rows). Thus, each pixel location (row in **X**) representing a particular feature in the multivariate image would produce a unique combination of score values. Sets of two or three score value

combinations can be represented as single points in 2-D or 3-D score plots, respectively. This would result in point clusters of similar score value combinations in score plots (e.g. t_1 - t_2 combinations) representing pixels from similar features in the multivariate image. For example, a feature represented by a group of pixels from a multi-wavelength image would have a specific combination of spectral characteristics which would map onto a unique region in the score plots thus producing a point cluster.

Because of the large number of pixels to be plotted there could be several overlaps of points in the score plots, producing various point clusters. This overlap may be serious in some cases where one point on a score plot may represent tens or hundreds of pixels from a multivariate image. Unfortunately, if the score plots are drawn as simple black and white scatter plots there would be no way of visualizing the number of overlapping pixels at a particular point. The number of pixels represented by a single point in a score plot is called pixel density [Geladi et al., 1996]. An intensity matrix can be constructed which represents the number of pixels at each point in the score space. This can be accomplished by simply assigning an intensity value to a point depending upon the number of pixels represented by that point. Geladi et al. [1996] provide several schemes to construct an intensity matrix, which can then be viewed as a 256 x 256 intensity image. To distinguish between different intensity values, color coding schemes can be implemented so that different colors are assigned to different points on the score plots, depending upon the pixel densities. Geladi et al. [1996] also list some subjective problems that can arise in interpreting the score plots since there is no unique best color coding scheme.

A color coding scheme similar to Geladi et al. [1996] has been used to construct the score plots for the satellite multivariate image example described above. The intensity matrix is constructed as a 3D histogram with a grid of 256 x 256 bins. Each bin is filled with a bar giving the count of pixels belonging to that bin. This matrix is then color coded depending upon the number of pixels in each bin (i.e. pixel density). The color scheme used is presented in table 3.1. Cold colors (e.g. black) represent bins with a

low number of overlapping pixels, whereas hot colors (e.g. white) represent bins having the highest pixel density.

Table 3.1 Mapping of Intensity Ranges from 3D Histogram Intensity Matrix to Colors

| <i>Intensity Range</i> | <i>Assigned Color</i> |
|------------------------|-----------------------|
| 0 - 0 | Black |
| 1 - 1 | Brick Red |
| 2 - 2 | Bright Red |
| 3 - 6 | Orange |
| 7 - 15 | Burnt Orange |
| 16 - 33 | Ochre |
| 34 - 75 | Yellow |
| 76 - 150 | Metallic Gold |
| 151 - 255 | White |

Using the above color scheme, score plots of the satellite multivariate image example are displayed as color images with dimensions of 256 x 256 pixels. Since there are four latent variables of the satellite image example, a total of six unique score plot combinations are possible. Figure 3.6 illustrates a selection of four of these color coded score plots. The space defined by the score plots of a multivariate image is referred as the 'score space' in the MIA literature.

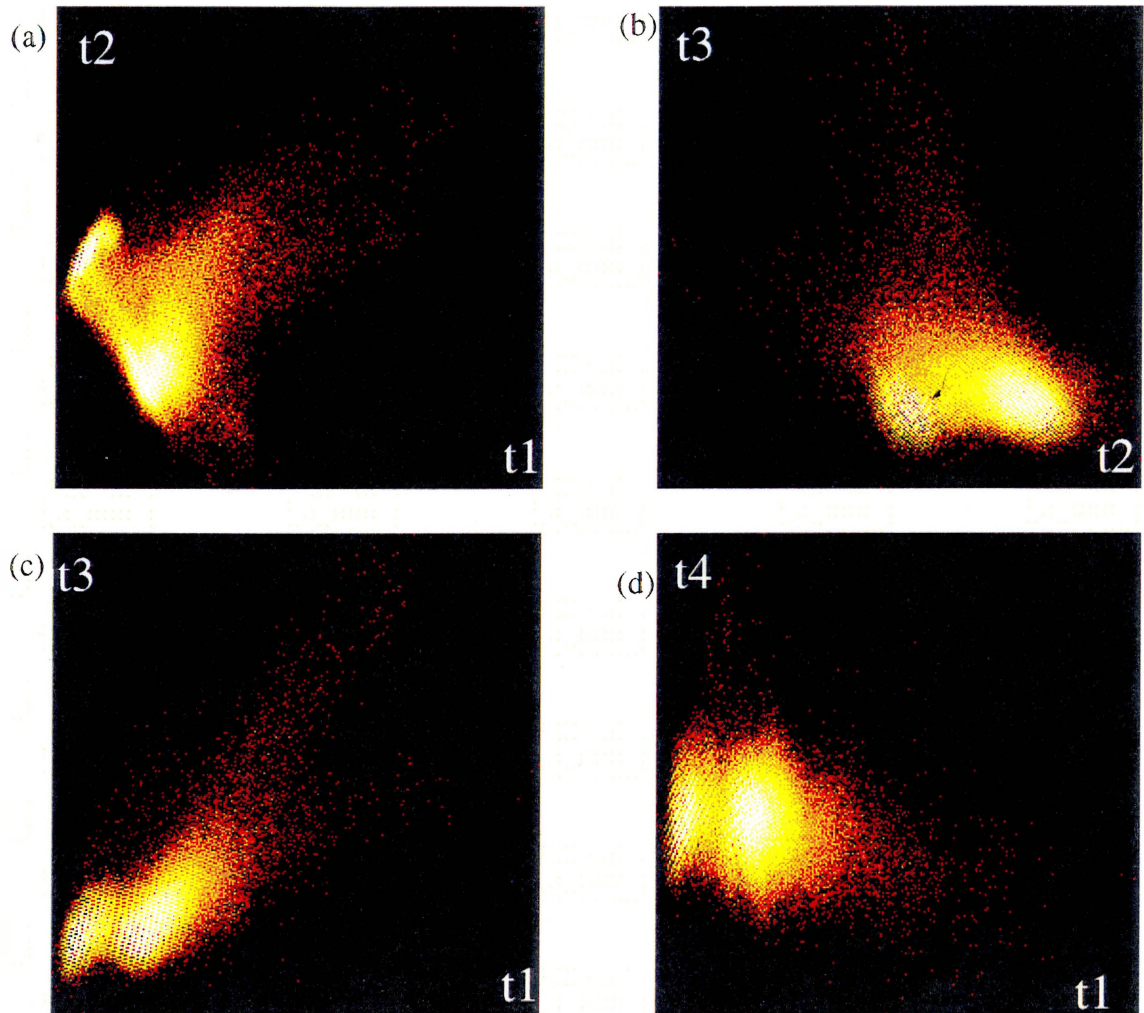


Figure 3.6 Color coded score plots of the satellite multivariate image: (a) $t_1 - t_2$, (b) $t_2 - t_3$, (c) $t_1 - t_3$, (d) $t_1 - t_4$

It is relatively easy to detect outlier pixels which are remote from the pixel clusters in the score space. It is also easy to detect high density clusters and the various pixel density gradients that exist both within and between clusters. However, all four score plots are not necessary for analyzing the satellite multivariate image. As discussed above, the first two latent variables explain majority of the sum of squares in the reorganized two-way array \mathbf{X} . Score plots that contain latent variables 3 and 4 can be discarded from any further analyses since these variables contain mostly noise with very

little structure. As a result, only the $t_1 - t_2$ score plot will be used for further analyses of the satellite multivariate image. A closer observation of the $t_1 - t_2$ score plot {figure 3.6 (a)} reveals two major dense classes of pixels which are separated by a prominent area of between-class pixels. There is also an upward slanting area of outlier pixels which might not belong to either of the classes defined by the two main point clusters. As mentioned earlier, pixels belonging to similar features in the multivariate image would exhibit similar spectral features. Therefore, similar feature pixels have comparable combinations of score values and result in point clusters in the score plot. This fact can be put to significant use in segmentation of features from the multivariate image through delineating pixel classes in the score plots. In effect, one can delineate a tentative data class corresponding to pixels having similar spectral fingerprints [Esbensen et al., 1989].

Pixel class delineation may be carried out in more than one way. Univariate delineation may be done by selecting a range of pixel intensities and highlighting only those pixels in the image that might belong to the selected range. In the multivariate case pixel class delineation is carried out by selecting an area in the score space and highlighting the corresponding pixels belonging to this area in the image space. The selected area in the score space is in fact a local model which is chosen to delineate a tentative class of pixel data from the rest. The procedure of selecting an area in the score space is called 'masking' in the MIA literature. As a result, multivariate pixel class delineation in MIA is carried out by masking point clusters (i.e. creating a local model), and highlighting the corresponding pixels in the image space. Since score plots can themselves be represented as 256 x 256 images, various sizes and shapes of masks can be selected by carrying out simple graphical operations on these score plot images. Using the '*roipoly*' (region of interest selection by polygon) command in the Image Processing Toolbox (v1.0b) of MATLAB[®] (v4.2c1) it is relatively easy to choose any arbitrary size and shape of polygon to highlight particular pixels in the score space. The use of polygons is a big advantage over using rectangular, or other 'fixed geometrical shape' pixel classifiers since polygons can have any desired shape and size [Geladi et al., 1996].

Once masked, the pixels contained within the polygon can be isolated and projected back to the image space. Another advantage of delineating a pixel class in the score space through polygon masking lies in the ability to choose all pixels with similar spectral features, i.e. belonging to a particular class, no matter how they are distributed in the scene space.

This procedure of masking point clusters and outlier pixels in the score space and highlighting the chosen pixels in the image space forms the backbone of the MIA off-line feature extraction strategy. To successfully delineate a class of pixels it becomes imperative to study both the score and image spaces simultaneously. The ability to toggle between the image space and score space is quite fast and easy since both spaces are represented as images in MIA. As a result, it is relatively simple to interrogate the score space by masking several clusters in the score space, and projecting the masked pixels back to the image space. Figure 3.7 presents a detailed image of the $t_1 - t_2$ score plot for the satellite multivariate image. The maroon polygon covering half of the lower dense point cluster represents a manually applied class mask. The pixel class which has been masked in figure 3.7 is outlined in figure 3.8 which illustrates the highlighted pixels (in white) overlaid on a false color image of the first latent variable T_1 . From figure 3.8 it is evident that the class of pixels masked in the score space belongs to major roadways, paved areas, and building tops in and around the city. Since all the highlighted pixels have similar spectral combinations, they map into the region masked by the polygon in the score space.

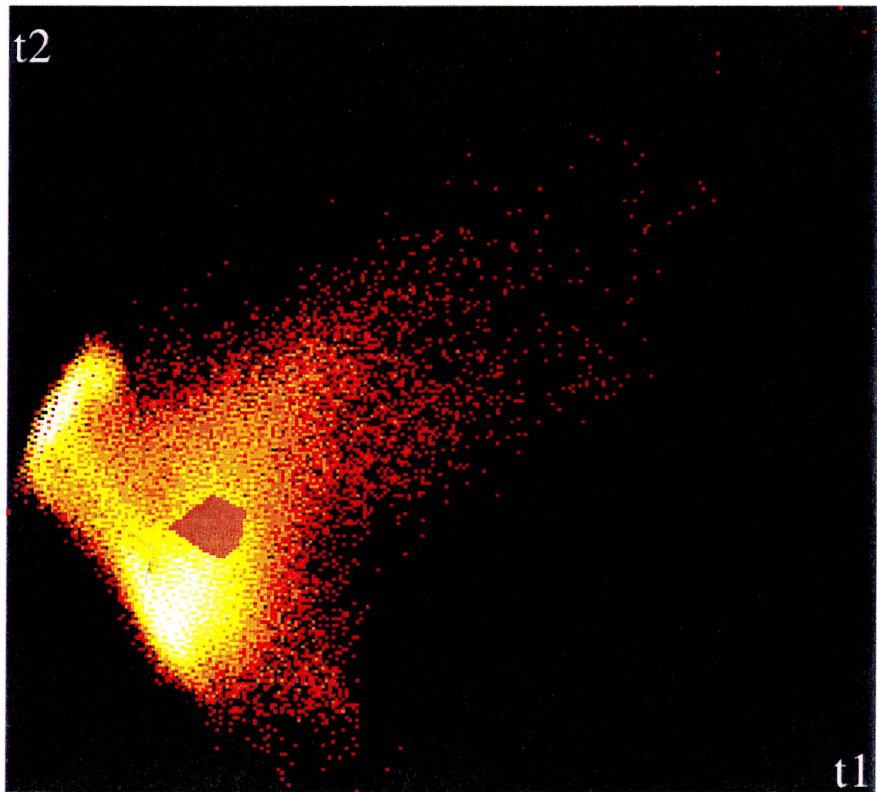


Figure 3.7 Detailed $t_1 - t_2$ score plot of the satellite multivariate image with a class-masking of the upper-half of lower dense pixel cluster (masked in maroon)

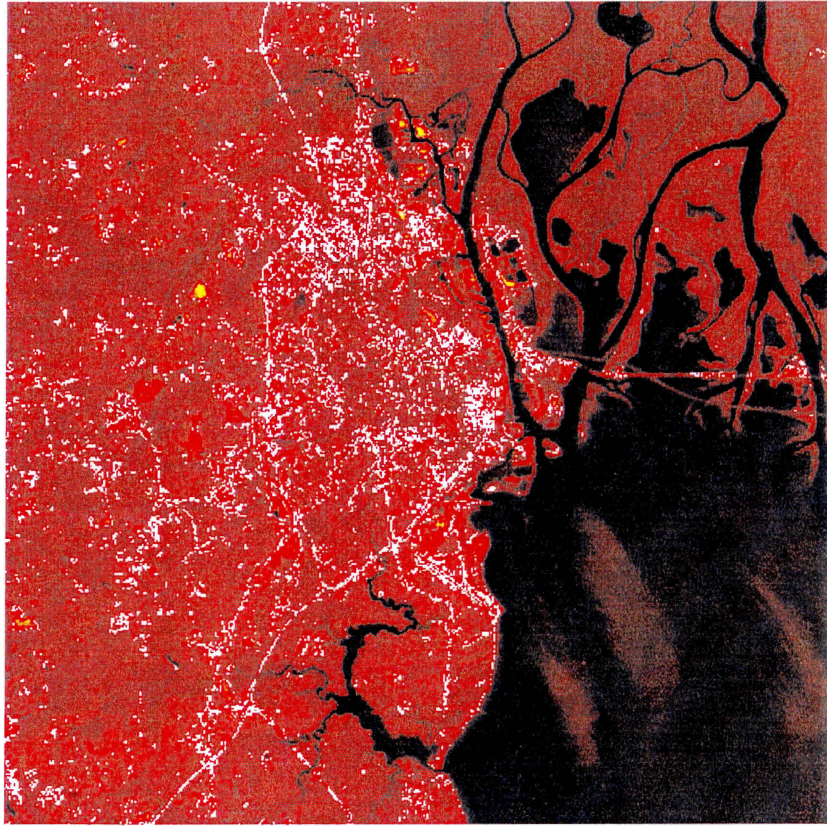


Figure 3.8 Overlay of T_1 image with highlighted pixels from class outlined in figure 3.7

The duality between the above two 'projections' (score space / image space) is complementary throughout any MIA application. No score space class can be considered without its complementary image space layout [Esbensen et al., 1989]. Repeated use of the masking/highlighting procedure with different polygon masks covering various point clusters in the score space extracts the relationships between unique features in the image space, and their corresponding score space clusters. In this way, a *signature* of every feature existing in the image space (regardless of its subtlety or spatial location) is collected in the full score space. Due to the ability to switch easily between score and image spaces, MIA can also be employed as a reverse mode image analysis tool. Specific pixels belonging to features of interest in the image space can be highlighted in the score space to determine the region which represents their corresponding score combinations. The area surrounding the highlighted score points can then be masked using a reasonably

sized polygon. As a result, subtle features that are subjectively difficult to identify in the image space may easily be identified using the reverse mode application of MIA. The polygon mask used in figure 3.7 was developed using this methodology since it is difficult to identify all the paved areas feature pixels from a visual inspection of the satellite image. Geladi et al. [1996] list several other modes of MIA that employ the use of the image space / score space relationship.

The user friendly and visually based feature extraction aspect of MIA has the potential of being applied in various fields that acquire multivariate image data. However, there are no set restrictions for the input data to be visual in nature. Image synthesis operations (discussed in traditional image processing techniques) that produce computationally designed images like Magnetic Resonance Imaging (MRI) / Ultrasonic / Positron Emission Tomography (PET) three-dimensional image data can be considered valid multivariate input data for extracting relevant information through MIA. Magnetic resonance images have been condensed using PCA to reduce the number of manifest variables and project the most significant information onto latent variable images [Grahm et al., 1989]. Various tissue classes have been successfully segmented using MIA feature extraction techniques from MRI multivariate images [Geladi et al., 1996]. Exploratory image analysis has been applied to several multivariate microscopic image data sets in chemistry [Geladi et al., 1986; Geladi et al., 1989; Geladi et al., 1992]. MIA has also been successfully applied to analyze and segment the paint used on an old piece of chinaware through a multispectral microscopic image [Geladi et al., 1994; Geladi et al., 1996].

Besides using PCA to extract features from multivariate image data, MIA employs several regression based latent variable methods to perform various tasks. Multivariate image regression (MIR) is described as the building of regression models between blocks of 'independent' (X) and 'dependent' (Y) multivariate image data and using these models to predict the dependent image block [Geladi et al., 1991; Esbensen et al., 1992]. The main ideas of MIR have been demonstrated by Geladi et al. [1996]

through simple examples. MIR has been successfully used in conjunction with MIA to monitor growth, progression, and effects of a treatment in a clinical study through MRI multivariate images [Grahn et al., 1992]. Partial Least Squares (PLS) regression has also been applied between MRI multivariate images of a laboratory rat and a set of manually constructed binary images (dependent block) to construct models that would predict the dependent block from new MRI images. This enhanced the contrast of the predicted image to better discriminate tissue types of the sample [Grahn et al., 1989]. An excellent literature review of MIR has been provided by Geladi et al. [1996]. The interested reader is encouraged to consult this reference for further details regarding various applications of MIA and MIR.

As described at the beginning of this chapter, the main focus of this research is the application of multivariate digital image analysis techniques to time-varying processes. The satellite image application of MIA for the purpose of feature extraction provides the basis for techniques of fault detection and object classification. An example may include the use of these ideas in processes which require off-line template matching or pattern recognition of machine produced parts from multispectral images collected via visual sensors. However, most industrial processes are on-line and progress in real-time. Several on-line multivariate imaging sensors can collect thousands of images to monitor these industrial processes on a daily basis. As a result, it becomes important to be able to detect faults and monitor industrial processes by analyzing this enormous multivariate image data on-line, and in real-time. Thus, there exists a need for an on-line application of the feature extraction aspect of MIA that can extract important information from a large amount of multivariate image data, and produce the desired results quickly and easily. The following section introduces an on-line multivariate image analysis technique that has the potential for monitoring industrial processes, and for detecting and isolating faults in real-time.

3.2 Real-Time Monitoring Using MIA

A major portion of the industrial processes that are, or will be, equipped with on-line imaging sensors are time-varying in nature. Traditionally, imaging sensors (e.g. movie cameras) have been used to monitor industrial processes on a continuous basis, thus producing movie segments. In various instances the most common mode of process monitoring using these imaging sensors has been a simple observation of the movie segments by operators to determine adequate operation of the process. However, this method of process monitoring has several drawbacks. First, it is entirely subjective in nature and as a result, is totally dependent on the judgment of the particular operator. Second, it is extremely difficult to detect subtle changes in the process by visual inspection of images.

With the advent of computers and various other technical breakthroughs, the imaging sensors used to monitor industrial processes are moving towards gathering discrete imaging data with information in wide bands of spectra. The image acquisition speeds of these sensors is steadily increasing due to increasing computational power. As a result, the amount of imaging data available to monitor industrial processes is steadily climbing. Although several traditional image processing techniques that can extract useful information from these images exist, they are not widely used for monitoring industrial processes. This is because of the long computation times required by these techniques to process the large amount of (multivariate) imaging data. Due to the time constraint, several industrial process are forced to use simple image processing techniques (e.g. thresholding, area analysis etc.) to extract only basic information. This may result in vital information regarding the industrial processes being lost.

In the previous section, a relatively new image analysis technique was reviewed which used latent variable statistical methods to decompose the large data arrays from a multivariate image into a reduced-dimensional sub-space. This sub-space (which is termed 'score space') serves as a data reducer while retaining all the important

information from the original image. The score space also reorganizes the information into a more usable format from which one could easily extract various features. Each score point in the score space could be associated with a corresponding feature pixel or groups of pixels in the multivariate image. Upon interrogating the score space by masking the corresponding score point clusters and highlighting the respective feature pixels (covered by the mask) in the image space, specific information on certain features could be extracted.

This technique of feature extraction from multivariate images can be extended to a sequence of time-varying multivariate images (called '*multi-temporal multivariate images*' [Geladi et al., 1996]) by simultaneously updating the score space every time a new multivariate image is acquired. This new multivariate image would contain updated information regarding the time-varying process. Upon decomposing the new image into its score space, the corresponding score combinations would be updated, thus leading to new pixel densities of score point clusters. This leads to the idea of monitoring the changing score point cluster intensities to infer the corresponding changes in the number of feature pixels in the image space. As a result, one can monitor features of interest in a sequence of multi-temporal multivariate images by monitoring the pixel densities of the corresponding point clusters in the updated score space. Tolerance limits could be set on the pixel densities of certain point clusters in the score space. Violation of these limits would indicate a significant change in the number of pixels belonging to the corresponding feature of interest in the image space. Since MIA allows one to easily toggle between score and image spaces, once a violation of the pixel density tolerance in the score space is discovered, one could switch to the image space to determine the spatial locations of pixels. This feature monitoring strategy is ideally suited for monitoring time-varying industrial processes for the purposes of fault detection and on-line feature monitoring. Furthermore, this process monitoring strategy provides a more objective analysis to determine the occurrence of faults, or of time-dependent variations in the process.

Some industrial processes that progress at high speeds require fast result acquisition to be successfully monitored. As a result, for the satisfactory application of MIA to monitor industrial processes there exists a need to develop a fast algorithm which can extract information from the multivariate image sequence in real-time. An extension to the traditional MIA feature extraction technique is introduced in the following section with the help of an example from the remote sensing field. This technique may be used in real-time to monitor features from a sequence of multi-temporal multivariate images. In the example it is shown that this technique can successfully monitor landscape features of interest from a sequence of time-varying LANDSAT (MSS) satellite images.

3.2.1 Extension of MIA Technique for Real-Time Monitoring

This section introduces an on-line extension to the traditional MIA feature extraction technique. This technique is mainly aimed at monitoring features of interest from sequences of multi-temporal multivariate images for fault detection and process monitoring. In order to develop a fast algorithm that can be used to monitor time-varying processes in real-time, one needs the ability to quickly distinguish between normal and abnormal operations of the process. This is usually accomplished with the help of a pre-developed training model that accounts for normal variations in the process. Once the process is operational, the training model can be used to calculate the updated score space to determine if the process is functioning within the bounds of normal variation (which is pre-defined by the training model). As soon as a fault occurs within the process the operator is alarmed and the model can be interrogated to show where, in the image space, the fault(s) have occurred. The development of a good and representative training model is an imperative first step in developing an on-line monitoring scheme.

In the area of MIA, to monitor sequences of multivariate images from a time-varying process, one needs to develop a training model in the reduced-dimensional latent variable space. This is accomplished by a latent variable decomposition using multivariate statistical methods (e.g. Multi-way PCA / PCR / PLS) of a training

multivariate image. This multivariate image must be representative of all the typical features from the process, i.e. it must contain at least a small sample of all features that one wishes to monitor in the future. Furthermore, the training image should have the same dimensions as each multivariate image from the sequence of process images. This is because the score space of the training image should have the same number of score points as compared to the score spaces of subsequent process images. The development of a PCA training model using an unfolded training multivariate image \mathbf{X}_t having dimensions of $(256 \cdot 256) \times 4$ is illustrated in figure 3.9.

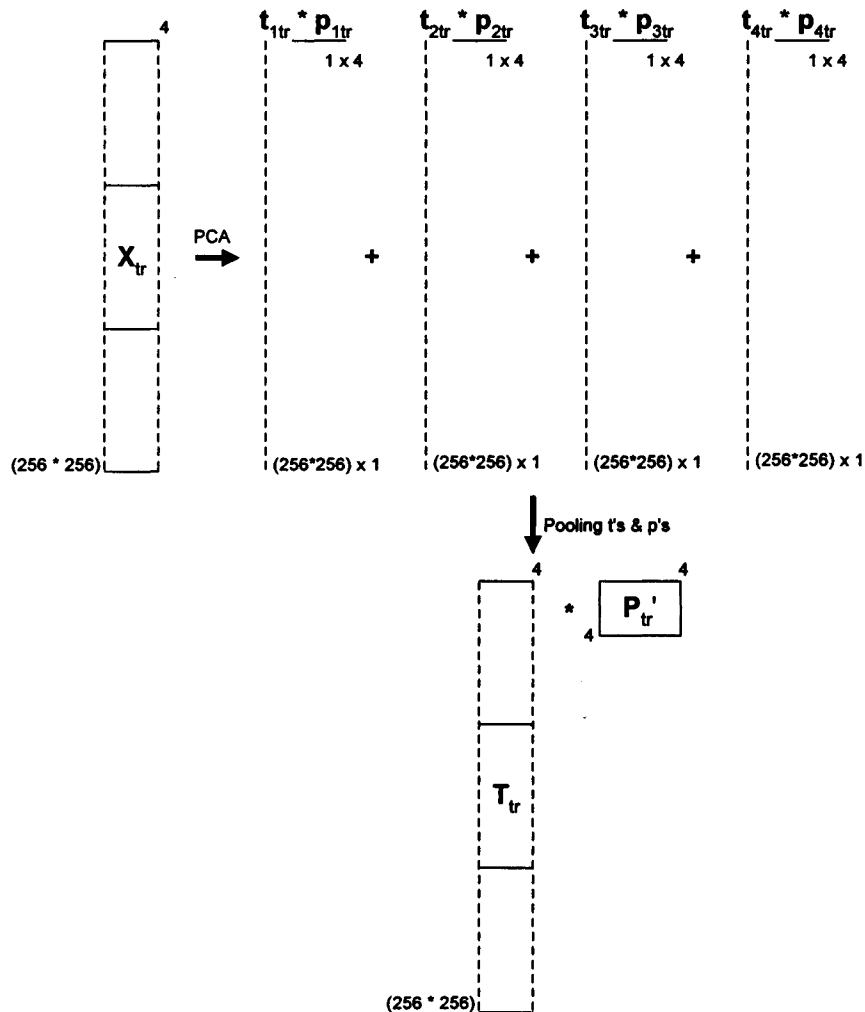


Figure 3.9 Latent variable decomposition of an unfolded training multivariate image to develop a PCA training model

In the above figure, X_{tr} is decomposed into its latent variable score matrix T_{tr} and loading matrix P_{tr}' . Together, the score and loading matrices of a training multivariate image form the training model which accounts for all the typical features in the subsequent multivariate images from the sequence of images used to monitor the particular process.

The score matrix of the training multivariate image can be used to serve a dual purpose similar to traditional MIA techniques. First, the score vectors (columns of T_{tr}) can themselves be viewed as images upon folding back to $(256 \times 256$ in figure 3.9) two-

way arrays; thus compressing information from the training image into a reduced space. Second, the score vectors can be plotted against each other as scatter plots (score space) with a color coding scheme to determine the pixel density of the score point clusters. As explained in the traditional MIA overview, the score space of a multivariate image is crucial for extracting features of interest from a multivariate image. This can be accomplished by interrogating the score space through manually applied local area masks on the score point clusters, and highlighting the corresponding feature pixels in the image space. One can toggle back and forth between score and image spaces to determine a suitable size and shape of a particular score space mask that corresponds to pixels belonging to the feature of interest in the image space. Repeated use of this strategy on the score space of the training multivariate image produces several manually shaped customized masks that correspond to various features of interest in the training multivariate image. This iterative procedure provides a calibration of the PCA training model that defines the relationship of each score cluster in the score space with its respective feature in the image space.

Once the score space has been fully interrogated, the calibration of the PCA training model is complete. Upon completion of the calibration step, features of interest can be monitored by depicting a time-varying process as a sequence of multivariate images. Each multivariate image in this sequence carries vital information regarding the status of the process. Using the training PCA model, score spaces can be constructed in real-time for each of these multivariate images. Every time a new multivariate image is provided by the multispectral sensors, it can be unfolded into a corresponding two-way array \mathbf{X}_{test} . Using the training PCA loadings \mathbf{P}_{tr} one can quickly calculate the score matrix \mathbf{T}_{test} for the new multivariate image:

$$\mathbf{T}_{\text{test}} = \mathbf{X}_{\text{test}} \cdot \mathbf{P}_{\text{tr}} \quad 3.4$$

To implement the MIA monitoring technique in real-time one needs to use the statistically significant number of loading vectors (i.e. columns of \mathbf{P}_{tr}) in the above equation. This produced a score matrix \mathbf{T}_{test} containing statistically significant score

vectors (i.e. columns). The score space for every new multivariate image in the sequence is produced by plotting the significant score vectors against each other. As explained earlier, this space can itself be represented as color coded intensity images. The above procedure is repeated for each new multivariate image from the multispectral sensors monitoring the time-varying process. As a result, a corresponding sequence of score plots, which represents the score space of the sequence of multivariate images, is developed. The essence of monitoring time-varying processes in real-time using MIA lies in this sequence of score space images. Instead of monitoring the image space for changing features one needs to monitor the corresponding score point cluster intensities in the score space. This is accomplished by monitoring the changing pixel densities under the pre-determined local area masks in the calibration step of the PCA training model. As the score space is updated with each newly acquired multivariate image, the pixel densities represented by the number of feature pixels in a score space local area are also updated. As a result, the increase or decrease of a specific feature in a time-varying process is easily captured by the change in the pixel density of the corresponding score point cluster in the updating score space.

Known faults that commonly occur in a time-varying process are used in the calibration step of the PCA training model to select masks in the score space for each fault. As a result, subsequent occurrences of these faults, while the process is operational, have their scores map onto their respective score space locations; thus, increasing the pixel densities in those local areas. One simply needs to monitor the respective score point cluster intensities to infer the occurrence of a particular fault while the time-varying process is operational. Since MIA allows one to easily toggle between the score and image spaces, the exact spatial locations of these faults, and other features of interest, may quickly be determined in real-time. This strategy of process monitoring and fault detection provides an objective image analysis scheme since exact number of pixels belonging to the particular feature, or fault, may be counted. This is accomplished by keeping track of the number of score points that map underneath each local area mask of

interest as the score space is updated. Automatic alarms can be set which immediately flag upon violation of a pre-set tolerable number of pixels in the image space belonging to a particular fault or feature of interest. This could be implemented through pre-set limits on the pixel densities of the corresponding point clusters in the score space. As a result, as soon as an alarm is sounded the operator may switch back to the image space to determine the type, and exact spatial location, of the fault in the image space. Depending upon the type, severity, and location of the fault, the operator may either stop the process for rectification, or mark the particular image number from the sequence of multi-temporal multivariate images for rectification of the corresponding area at the end of the process. The severity and spatial location of the fault in the image space may also aid the operator in diagnosing an assignable cause for the problem.

The above ideas and concepts of fault detection and feature monitoring using MIA are applied to a sequence of multispectral satellite images in the following section. This example is used because it represents one of the most advanced real-time imaging systems currently being employed, and because it exhibits all of the characteristics that might be present in future real-time process imaging sensors.

3.2.2 Example of Feature Monitoring using MIA: Remote Sensing

A modified version of the LANDSAT (MSS) multispectral satellite image example (depicting Mobile, Alabama) introduced in the section used to overview traditional MIA techniques is used here to illustrate the main ideas of feature monitoring. Details regarding this multispectral image may be obtained from the overview section. The original 512 x 512 x 4 multivariate image has been modified into sections of 256 x 256 x 4 multivariate images to illustrate a sequence of multi-temporal multivariate images. This sequence could be gathered from a satellite acquiring snapshot multispectral images as it passes over a certain geographical region of the earth's surface. In this particular example, the sequence of 256 x 256 x 4 multivariate images represents

acquired snapshots of the city of Mobile, Alabama and its surrounding areas as the LANDSAT satellite moves in a north to south trajectory over this region.

As discussed above, it is extremely important to select a training image that contains representative samples of all the features of interest to build a good PCA training model. In the satellite image example there are four main features of interest that are chosen from various landscape features in and around Mobile, Alabama. Some of these features of interest are subjectively easy to detect, whereas some are not so obvious. The four main features of interest that are chosen to be monitored throughout the sequence of multivariate images, as the satellite moves over the region, are; (a) deep water areas, (b) shallow water areas, (c) major roads & paved areas, (d) golf courses. As a result, a good representative training multivariate image must contain ample pixels belonging to all of the above four features of interest. The center 256 x 256 x 4 pixel sub-section of the full scene (512 x 512 x 4 pixel) in the original multivariate image adequately meets this requirement. Figure 3.10 illustrates the 256 x 256 pixel sub-section chosen to be the training multivariate image which is outlined by a white box on a false color score (T_1) image of the full 512 x 512 pixel scene.

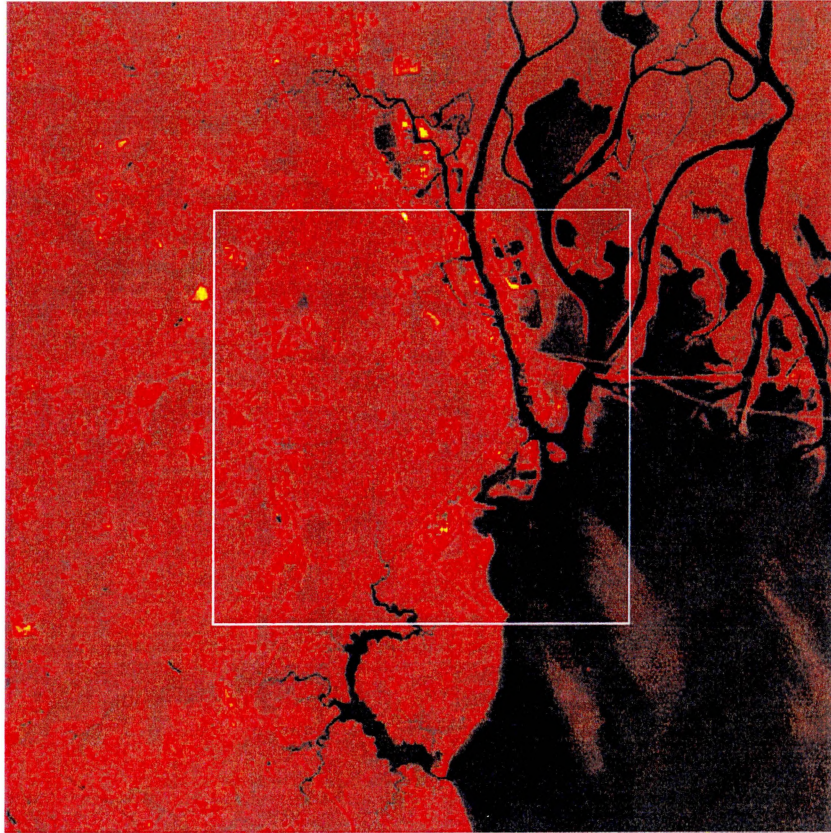


Figure 3.10 Training image (outlined by white box) chosen from full scene for calibration of PCA training model

The development of a PCA training model from a training multivariate image has been described in the previous section and illustrated in figure 3.9. Using this procedure on the unfolded $65,536 \times 4$ two-way array of the training image \mathbf{X}_t a PCA training model is developed. The loading matrix is calculated from SVD of the cross-product matrix ($\mathbf{X}_t' \mathbf{X}_t$):

$$\mathbf{P}_t = \begin{bmatrix} 0.369 & -0.510 & 0.769 & -0.111 \\ 0.434 & -0.585 & -0.535 & 0.428 \\ 0.604 & 0.172 & -0.280 & -0.727 \\ 0.558 & 0.606 & 0.210 & 0.526 \end{bmatrix}$$

The first two principal components account for the majority of the sum of squares in this multivariate image. Thus, only the first two loading vectors (columns of \mathbf{P}_t) are used to

calculate the significant score matrix \mathbf{T}_r ($65,536 \times 2$) using equation 3.2. The two score vectors (columns of \mathbf{T}_r) are folded back into 256×256 arrays to be viewed as images. The unfolded score vectors are also plotted against each other to form a t_1 - t_2 scatter plot. The reduced score space of the training multivariate image defined by the t_1 - t_2 scatter plot has been illustrated in figure 3.11 as a 256×256 color coded intensity image. Each pixel in this image is represented as an intensity value that is determined by the pixel density. The same pixel intensity coloring scheme from table 3.1 has been adopted in this plot to illustrate the pixel densities of the score point clusters.

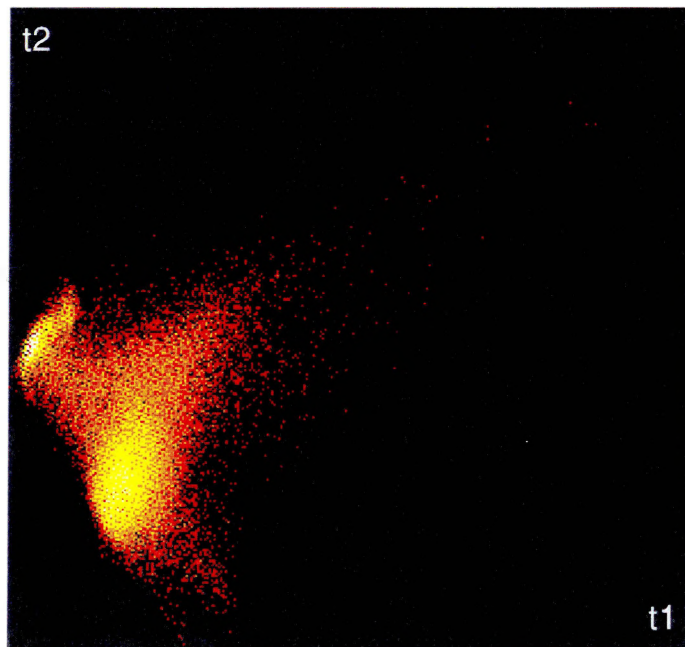


Figure 3.11 Detailed t_1 - t_2 score plot of the training multivariate image with color coded pixel intensities

The PCA training model is calibrated by masking various point clusters with customized local area score masks. The sizes and shapes of these masks are determined by using the previously discussed iterative procedure of feature extraction from multivariate images. In this case the features of interest were known before hand. As a result, the reverse mode of MIA is used to determine the spatial locations of selected

feature pixels in the score space. Once the score points are highlighted, manually shaped masks are applied such that the areas surrounding these scores are covered. All pixels that are covered by these masks are highlighted in the image space to determine if the particular masks adequately segmented the features of interest. This procedure is repeated until an adequate number of pixels belonging to the above four features of interest are highlighted in the image space. The resulting score space masks that adequately represent the four features of interest in the image space have been highlighted in figure 3.12.

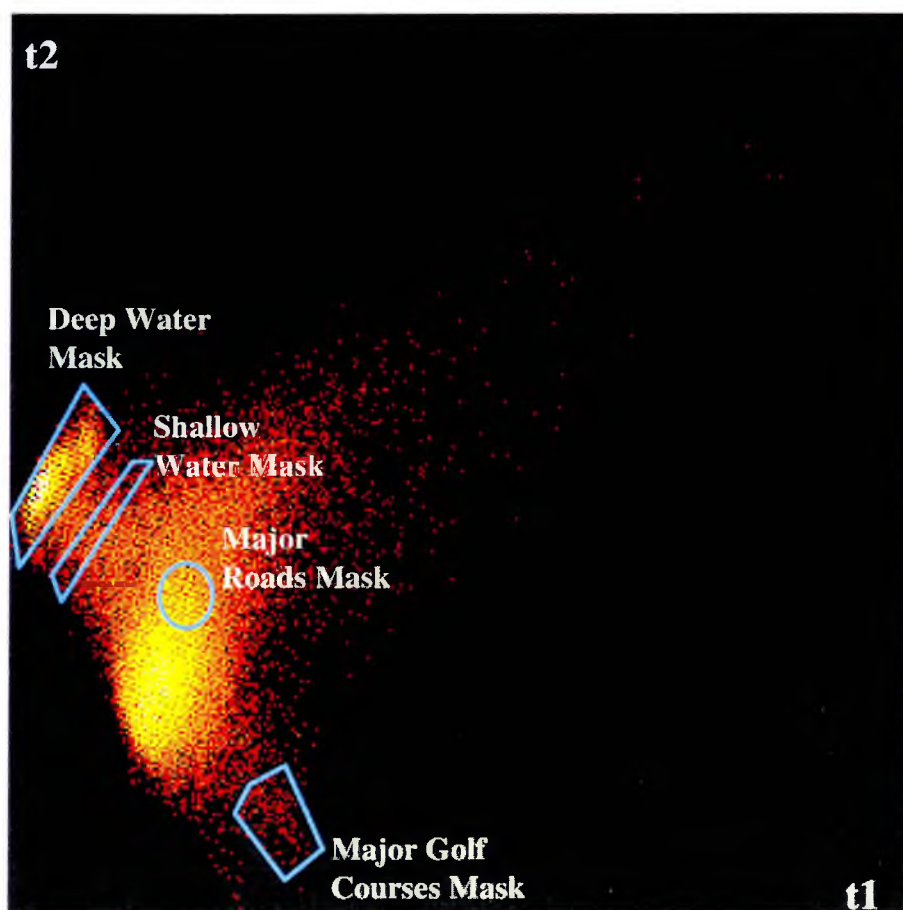


Figure 3.12 Detailed t_1 - t_2 score plot of training image with highlighted masks that adequately represent the four features of interest

Using the above four customized local area masks, along with the reduced loading matrix P_r , the calibration of the PCA training model is complete. This model is ready to monitor the four features of interest from a sequence of multivariate satellite images as it gathers $256 \times 256 \times 4$ scans while moving over the Mobile, Alabama region.

To emulate a moving satellite gathering scans of the earth surface the full $512 \times 512 \times 4$ pixel multivariate image is divided into scans having dimensions of $256 \times 256 \times 4$ pixels each. First, the full scene is divided into two halves (i.e. columns 1-256 & 257-512). The first half (i.e. columns 1-256) of the full scene is further divided into various sub-sections. The second half (i.e. columns 257-512) is not used in this example since majority of the pixels in this half only belong to deep water areas. The first half of the full scene is divided into 9 equal sized scans of 256×256 pixels in all four spectral bands. Since the LANDSAT satellite moves in a north to south trajectory while gathering images of the earth surface, the first $256 \times 256 \times 4$ scan is placed at the top of the first half image. Each successive scan is gathered as the satellite moves south in a certain time which is represented as 32 rows. As a result, the second $256 \times 256 \times 4$ scan spans rows 33 to 288 whereas the ninth scan spans rows 257 to 512 in the first half of the full scene. Using these nine scans a north to south moving LANDSAT satellite is emulated that gathers intermittent multispectral images as it passes over the region of Mobile, Alabama.

The 9 input multivariate images are decomposed into their score spaces with the help of the PCA training model. Each $256 \times 256 \times 4$ pixel multivariate scan image is rearranged into a $65,536 \times 4$ two-way array X_{test} and multiplied with the (reduced) training loading matrix P_r to produce the corresponding score matrix T_{test} . These score vectors are rearranged into 256×256 arrays and viewed as score images, as well as plotted against each other as color coded scatter plot images ($256 \cdot 256$ pixels) using the same coding scheme from table 3.1. As a result of the decomposition, 9 t_1 - t_2 score plot images can be plotted. Figure 3.13 illustrates three (scans 1, 5, 9) of the 9 scan images (outlined by white boxes) along with their corresponding t_1 - t_2 color coded score plot images.

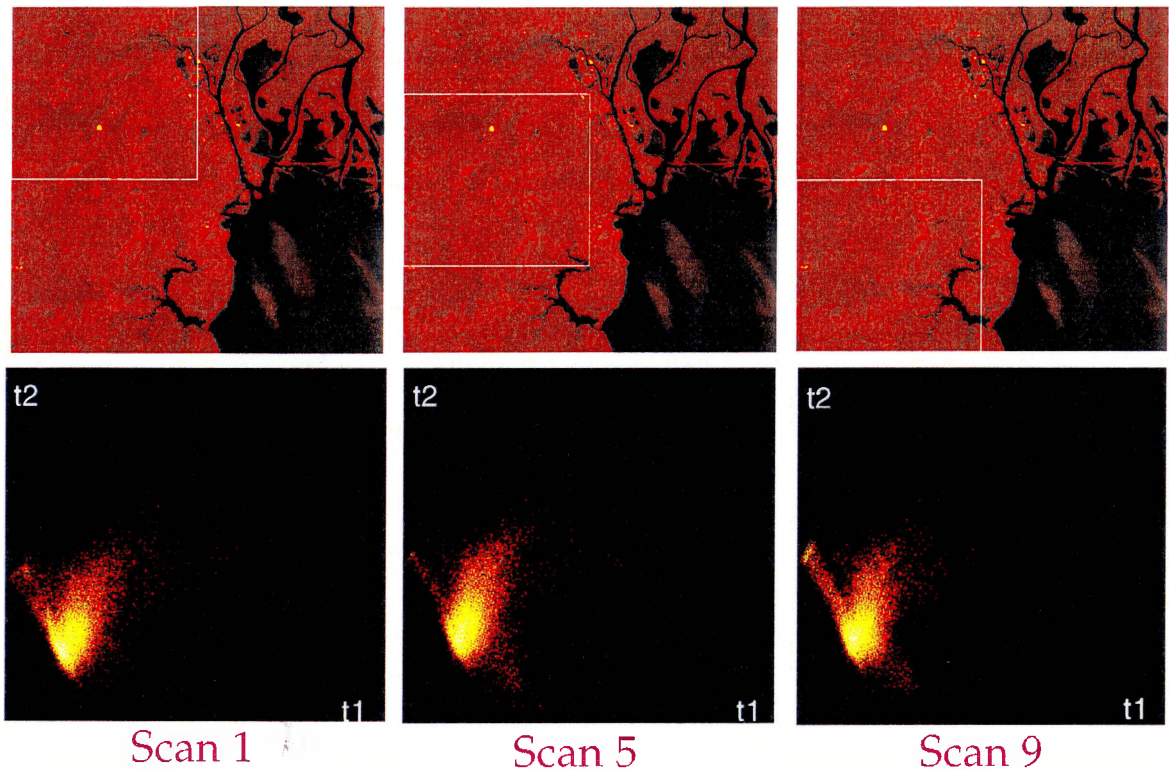


Figure 3.13 Three of nine (scan 1, 5, 9) input images outlined as white boxes on false color T_1 full scene score image with corresponding color coded t_1 - t_2 score plot images

Upon examining the 3 scan images in this figure, it can be seen that scan #1 captures regions that are mainly north of the city; the majority of the city areas are captured by scan #5; finally, scan #9 captures areas that are south of the city. As a result, one can gather some subjective impressions regarding the four features of interest upon observing these scan images. For example, it is obvious that scan #9 contains more pixels that belong to deep and shallow water areas as compared to scan #5. Similarly, scan #5 would be expected to have the most number of pixels belonging to paved areas since this image captures majority of the city. However, subtle features like golf courses would be more difficult to identify with a visual observation of these image space scans. As a result, a more objective feature monitoring scheme is necessary to detect the more subtle features

of interest. This scheme is also necessary to gather more precise information regarding the number of pixels belonging to the obvious features of interest.

As discussed in the previous section, one can monitor features of interest from a sequence of multi-temporal multivariate images by monitoring the pixel densities of the corresponding point clusters in the score space. This strategy is used to monitor the four features of interest through scans 1 to 9 in this example as the satellite moves over the region. The pixel densities of the corresponding four point clusters in the nine score space images are monitored to determine the exact number of pixels belonging to these four features of interest for each satellite scan. This is accomplished by counting the number of feature pixels for each scan that mapped underneath all of the four customized local area masks (figure 3.12). The total numbers of pixels belonging to each of the four features is recorded for all nine scan images. The exact spatial locations of these feature pixels in the image space are obtained by highlighting them in the score images upon switching back to the image space. This procedure is followed for each of the four features of interest through all nine satellite scan images. The resulting feature pixel counts and highlighted spatial locations of these pixels in the image space are illustrated in figures 3.14 - 3.17. The score images used to highlight the spatial locations of pixels belonging to the four features of interest are false color images of the first score vectors T_1 in each of these figures.

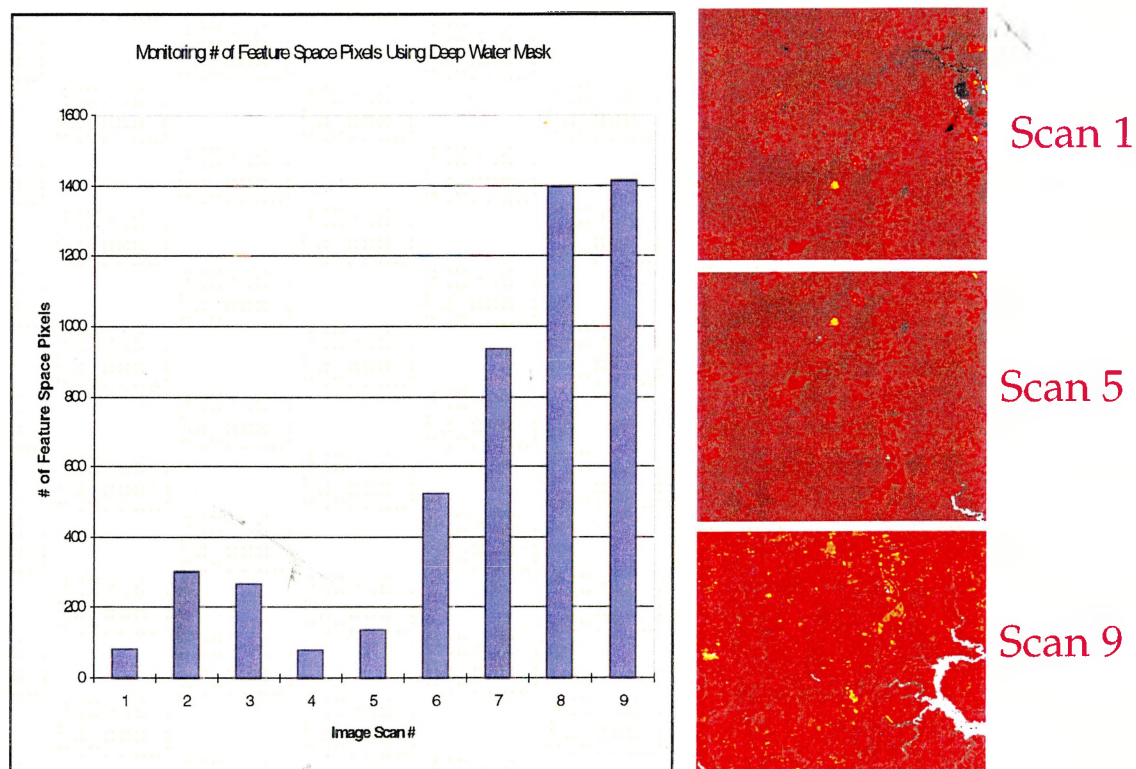


Figure 3.14 Bar chart of total number of pixels belonging to deep water areas through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9

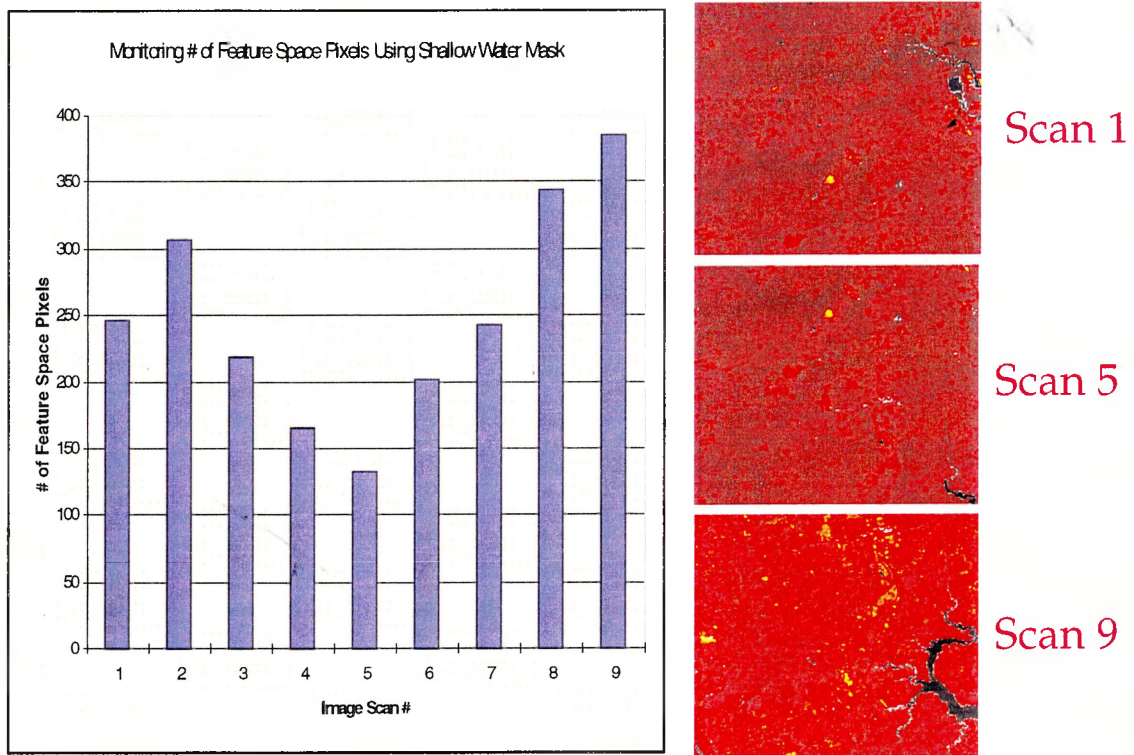


Figure 3.15 Bar chart of total number of pixels belonging to shallow water areas through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9

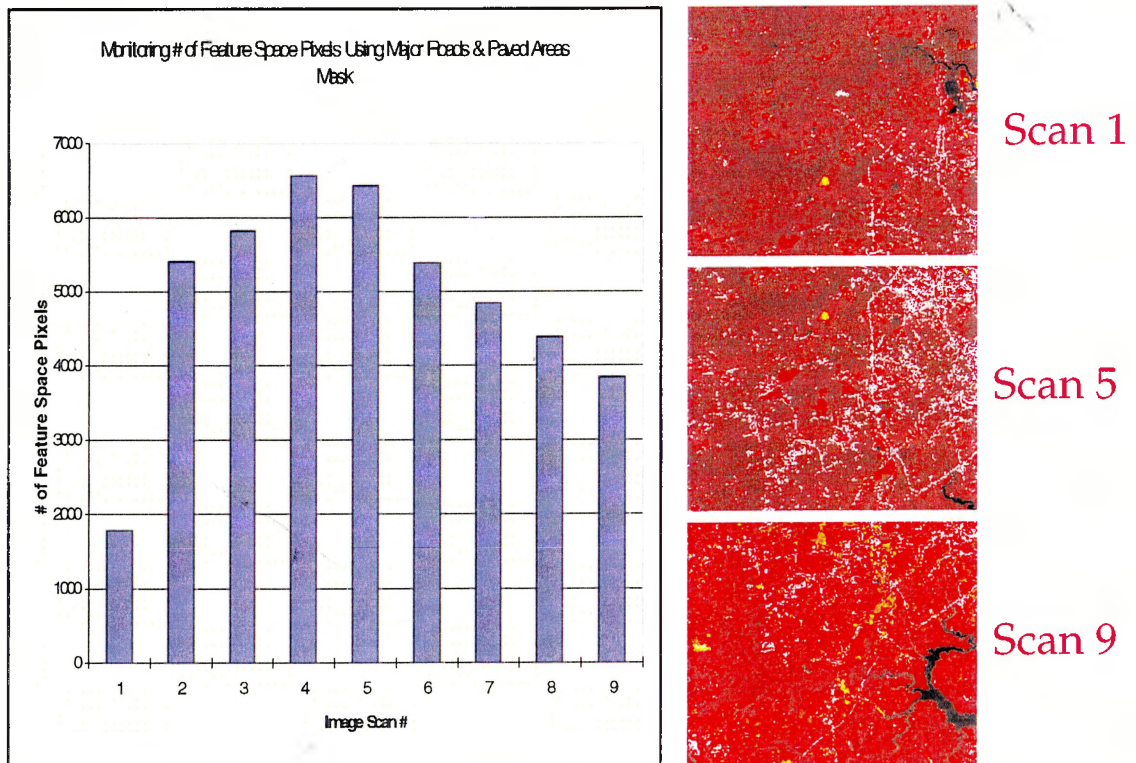


Figure 3.16 Bar chart of total number of pixels belonging to major roads & paved areas through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9

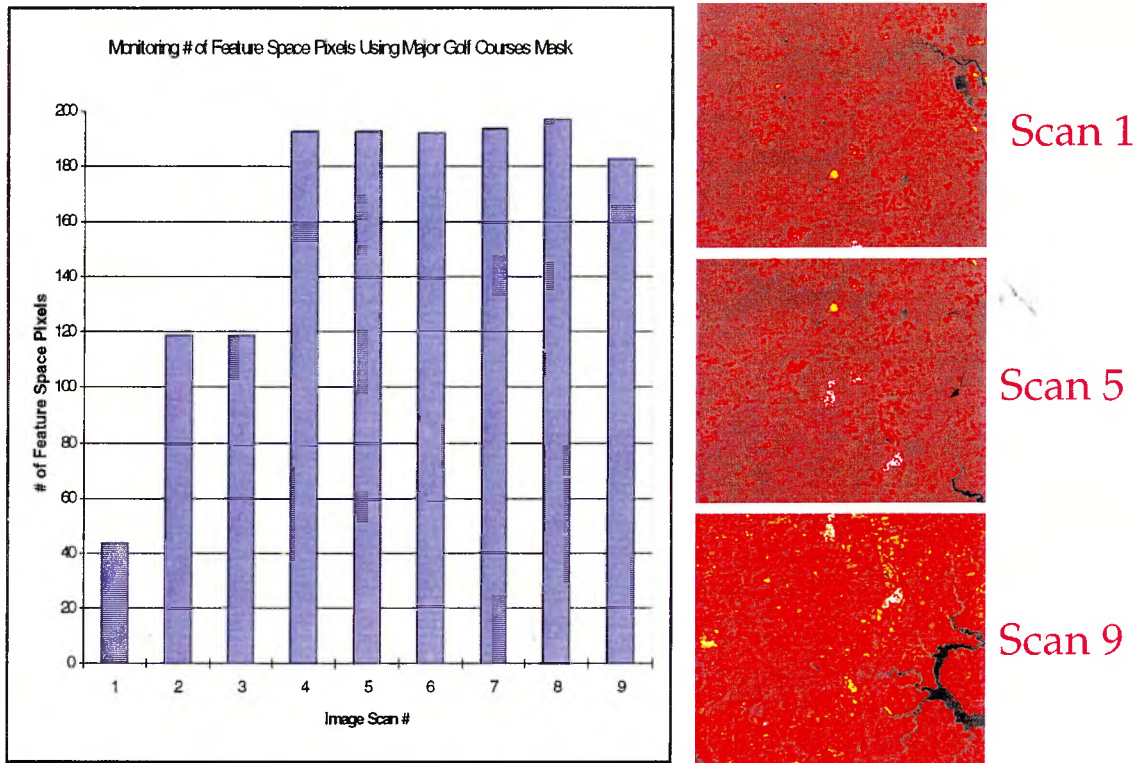


Figure 3.17 Bar chart of total number of pixels belonging to major golf courses through scans 1 - 9 with highlighted spatial locations of these feature pixels in the image space for scans 1, 5, and 9

The bar charts through figures 3.14-3.17 provide an exact number of pixels belonging to each of the four features of interest for all nine scan images. As a result, these charts give a more precise measure of the severity of all four features throughout the sequence of 9 multivariate images. The results contained in these charts strongly agree with those gained by a subjective analysis of the 9 scan images. The deep and shallow water pixels increase as the satellite moves to the south of the city (scans 8 & 9) whereas the paved area pixels are more pronounced in scans over the city (scans 4 & 5). There were a total of three major golf courses that the satellite captured as it moved over the region represented by the nine scans. These three golf courses are highlighted in scan 5 of figure 3.17. From the corresponding bar chart in this figure it is obvious that there are three distinct levels of pixel counts representing three major golf courses through scans 1-9.

As a result, all four pre-chosen features of interest were adequately monitored through the sequence of 9 multivariate satellite images.

The above example is a successful application of the feature monitoring technique of MIA in the field of remote sensing. In order to apply this type of strategy on a time-varying industrial process it is extremely important to be able to process and display the results in a reasonable period of time. As a result, there is a need to gather data regarding the amount of time required for obtaining some of the results in the feature monitoring example described above. Table 3.2 provides average times (averaged over all 9 scans per feature of interest) required to perform three major tasks while monitoring the 9 multivariate image scans for the four features of interest. The total time required to obtain results for each feature is dependent on several factors including; size of the local area mask in the score space, number of pixels belonging to feature of interest per individual scan. It should be noted however, that for process monitoring purposes one only needs to count the number of points under the score plot masks corresponding to the faults, or features of interest. Thus, the third column in table 3.2 is the critical time as far as process monitoring is concerned. The simulations for the above example were performed on an IBM compatible Pentium 166 MHz personal computer with 64 MB of RAM running in a Windows® '95 environment. The software used for the actual simulations was MATLAB® v4.2c1.

Table 3.2 Average Times (over 9 scans) to Monitor 4 Features of Interest from Satellite Multivariate Image Sequence

| Feature of Interest | Time to Evaluate & Plot Scores (sec.) | Time to Count # Pixels Under Mask (sec.) | Time to Overlay Highlighted Pixels on T_1 Image (sec.) | Total Time (sec.) |
|---------------------------|---------------------------------------|--|--|-------------------|
| Deep Water Areas | 30.85 | 122.07 | 1.06 | 153.98 |
| Shallow Water Areas | 30.12 | 96.04 | 1.01 | 127.18 |
| Major Roads & Paved Areas | 30.69 | 65.57 | 1.39 | 97.66 |
| Major Golf Courses | 30.78 | 94.05 | 1.01 | 125.83 |

These average times vary between approximately 1.5 - 2.5 minutes using the above mentioned software and hardware. If desired, these average times may be considerably reduced upon using faster compiler software equipped with more efficient code. However, the above tabulated average times provide a benchmark for comparing future applications of feature monitoring schemes using MIA. Some industrial processes that gather visual data using on-line imaging sensors (e.g. steel rolling, paper production etc.) may require much faster cycle times between gathering images and calculating results. However, in many process situations, updating the monitoring plots every 1 - 2 minutes would be quite adequate for fault detection.

The information that is gathered from the techniques of feature extraction and monitoring using MIA can be vital for implementing feedback control schemes. After detecting faults in a process using the above MIA techniques, several designed identification experiments could be performed on industrial processes to determine how certain manipulatable variables (e.g. oxygen feedrate to a section of a furnace being monitored with multispectral imaging sensors) affect these features in specific regions of the image space. This knowledge can be used for implementation of feedback control

schemes like spatial control, or Statistical Process Control (SPC) for industrial processes being monitored using multispectral sensors.

3.3 Conclusions

This chapter investigated the use of a relatively new and powerful statistically based image analysis technique called Multivariate Image Analysis (MIA). This technique was used to extract the full information content from the large multivariate data being collected by multispectral sensors. Furthermore, this technique was extended for monitoring features, and detecting faults from sequences of multi-temporal multivariate satellite images. Ideas regarding both off-and-on-line monitoring and feedback control schemes using the information gathered were also discussed for potential industrial applications of MIA.

The main ideas behind multivariate image data compression and feature extraction were presented with the help of a brief literature review of the theory and applications of MIA in the first part of this chapter. This section introduced the concepts of decomposing the multivariate image into its reduced dimensional latent variable space. This space contained the information content from the multivariate image and was represented by the loading and score matrices. The importance of interrogating the score space to detect the spectral combinations of various feature pixels and delineating these features using local area masks have also been described in this section. These ideas were then enforced by presenting an example from the remote sensing field for the purposes of data reduction and landscape feature extraction from a LANDSAT (MSS) satellite image.

The ideas behind the relationship of the image and score space introduced in the overview section were further applied in the second part of this chapter. This section

introduced the concept of monitoring features from a sequence of multi-temporal multivariate images using MIA. The ideas of calibrating the latent variable training model were introduced using a training multivariate image and customized local area masks in the score space of this image. This training model was used to construct a corresponding sequence of multi-temporal score space scatter plots. These plots represented the spectral combinations of all the feature pixels in the sequence of time-varying multivariate images. Using the customized local area masks to monitor the score point cluster intensities, it was shown how to monitor features (or faults) from a sequence of multivariate images. These ideas have the potential of being applied on sequences of multivariate images gathered from multispectral sensors monitoring time-varying processes. An example featuring a modified version of the same LANDSAT (MSS) satellite image (used earlier in the chapter) was employed to illustrate the introduced ideas of feature monitoring. Four landscape features of interest were successfully monitored through a sequence of 9 multivariate images as the satellite moved over a fixed geographical region.

Finally, other applications of the MIA feature monitoring scheme could be devised. Several industries that use multispectral sensors for monitoring various time-varying processes can benefit immensely upon employing either off-or-on-line feature (fault) monitoring techniques using MIA alongside their existent monitoring schemes. Furthermore, these ideas can also be used to form the basis of an automated real-time monitoring scheme that produces both subjective, as well as detailed objective results with the click of a button. Using the information gathered from these techniques, along with carefully planned designs of experiments (to determine proper manipulatable variables), various feedback control schemes could be devised to automatically correct the faults in the processes without the need of a shutdown.

4. SUMMARY AND CONCLUSIONS

The field of digital image processing has lately gained significant attention due to the recent advancements in computational speeds and technically advanced visual sensors. As a result, this area is finding a growing number of applications that range from electron microscopy to space exploration. Since digital image processing is adaptable for application in a vast number of fields, it is clear that the future of various technological applications lies in visually based systems. One of the greatest potential areas for this field is in industrial inspection and monitoring. A growing number of industries are moving towards automatic vision systems to monitor a lot of their processes. Several customized digital image processing techniques are currently being used in industries to analyze thousands of images collected by imaging sensors monitoring their time-varying processes. However, these techniques are currently restricted to simple analysis operations (i.e. low-level image processing) due to short cycle times between image acquisition and the need for results, and due to the large amount of data collected.

The main goal of this thesis is to create an efficient link between the fields of digital image processing and industrial process monitoring with the help of latent variable statistical methods (Multi-way PCA / PLS). This is achieved upon using a class of recently devised image analysis techniques called Multivariate Image Analysis (MIA). These statistically based techniques form the basis of an on-line process monitoring scheme. Eventually, the future of this research is aimed at using the results, obtained from the real-time application of MIA, for devising on-line SPC and feedback control schemes for various industrial processes.

This project is the initial step towards devising image based industrial process feedback control schemes through the use of MIA. As a result, to develop a strong foundation for the possibility of further research in this topic, it was imperative to get a firm understanding of the basics of digital image analysis. This was accomplished by performing a brief literature review of some traditional image processing techniques in chapter 2. These techniques were also studied to gain insights into various operations that could be potentially used as image preprocessing prior to applying advanced MIA techniques. After defining various types of digital images, this chapter introduced the five main classes that form the domain of digital image processing (image enhancement, restoration, analysis, compression, & synthesis). The first three out of these five classes were discussed in detail. The last two classes were of no help in gaining a basic understanding of the common techniques of digital image processing.

Perhaps the most commonly used digital image processing techniques belong to the image enhancement and restoration classes. The main operations comprising image enhancement include image histogram equalization and image adjustment (input & output cropping, gamma correction) for improving the contrast of the image. Image restoration operations are mainly used to remove several types of noise from the image. Filtering in both the spatial and frequency domains is the main tool used for image restoration operations. The frequency domain representation of a digital image is obtained upon performing the 2D-FFT, and representing the magnitudes of the Fourier coefficients at various frequencies as pixel intensities. Almost all image processing tasks involve some sort of enhancement and restoration techniques for either improving the final appearance of the image, or preprocessing the image for subsequent image processing operations. These techniques are also useful in other image analysis operations that are not necessarily restricted to the field of image processing. One such application of these techniques that deals with image classification using 2D-FFT's as preprocessing of the images has been discussed in the following section of this chapter.

The third main class of digital image processing is image analysis. This class is mainly used in machine vision and image interpretation applications. The main operations that are a part of image analysis include image segmentation, morphological operations, and object classification & measurement. Image analysis operations are based on the definition that an image is made up of two main categories. The category that forms the elements of interest in an image is called the objects. The rest of the image is categorized as the background. Image segmentation operations are aimed at separating the objects from the background through various techniques. Thresholding is the primary tool for image segmentation through categorizing each pixel in the image as an object or background, depending upon its brightness value. Although various automatic thresholding algorithms exist, this operation is usually done manually with a trial-and-error procedure of trying various threshold brightness values until optimum (subjective) categorization of the objects and background is achieved.

However, image segmentation may not be enough to produce the final results. Often resegmentation of the segmented objects may be required to further split them into sub-parts, or unite the adjoining objects into one. These tasks are performed by (set theory based) morphological operations. These operations employ a manually shaped structuring element which is logically combined with each pixel in the image to produce an output image containing objects with particular patterns that are being sought. Adjoining objects could be separated into two distinct objects using erosion operations, whereas dilation could be used to fill the gaps between adjoining objects thus making them one. Furthermore, these objects could be reshaped back to their original sizes using opening and closing operations.

Upon completion of the initial segmentation and resegmentation steps, the resulting objects may be measured for various types of geometric properties. This step is the domain of object measurement & classification operations which are responsible for extracting quantitative information (e.g. object size, shape, orientation, spatial moments)

from the preprocessed images. This is the final step in a traditional image analysis application.

The classes of image compression and synthesis are not discussed in this report due to their little relevance towards the immediate goals of this thesis. However, several references have been provided throughout chapter 2 that may be consulted to gain a detailed knowledge of these classes of traditional image processing.

After laying down the basic foundations of this research through an introduction to various traditional image processing techniques, this thesis overviews the use of latent variable statistical methods to analyze digital images in chapter 3. This is performed by investigating the use of MIA on multivariate images for the purpose of extracting useful information that could be further used for process monitoring, and feedback control. As a result, chapter 3 attempts to lay down the main concepts of this research in two interrelated steps. First, this chapter provides an overview of the theory and applications of MIA to extract features of interest from multivariate images. Second, these concepts are further extended to introduce a new strategy of monitoring features of interest from a sequence of multivariate images depicting a time-varying process.

The first part of chapter 3 presents an overview of the (relatively new) technique of Multivariate Image Analysis (MIA) through a literature review of both its theory and applications. This is initiated with a basic introduction of a multivariate image, and the different ways of viewing the arranged data in these types of images. The data contained in a multivariate image are highly correlated at each pixel location in the image plane throughout the 'variable' images and among adjacent pixel locations. Latent variable statistical methods (e.g. Multi-way PCA / PLS) have been known to extract the information content from highly correlated data sets by reducing their dimensionality to a few orthogonal latent variables. Thus, MIA employs these statistical methods to both compress the data, and extract the information content from a multivariate digital image. This is performed by establishing a latent variable space that is defined by the product sums of score and loading vectors. The score space of a multivariate image is the main

building block of MIA since this space can be used to both view and analyze the condensed information content in a multivariate image. A score image is a rearranged score vector into a two-way data array having same dimensions as the image plane in the multivariate image. Through this image, the score values can be analyzed as pixel intensities. Since scores are linear combinations of the variable images throughout the pixel locations, features that are variable specific in a multivariate image are condensed in the score images.

The second way of analyzing the multivariate image is through score plots. These are scatter plots of two or three score vectors. The scatter plots contain a large number of points (one point for each image plane pixel) which form several overlapping point clusters representing pixels from similar features. As a result, to effectively display these clusters, the score plots are displayed as color coded images. Each pixel in these images represents colors dependent on the pixel density at the respective spatial location in the scatter plot. Pixel class delineation can be carried out in the score plots to segment the corresponding features from the multivariate image through local area masking of the score clusters, and highlighting the underlying score points as pixels in the score images. Thus, the score plots and the score images together form the backbone of the MIA feature extraction technique. No score plot can be considered without its corresponding score image, and vice versa. Since both score images and plots are represented as images, it becomes quite simple to toggle between the two spaces; thus making it easy to interrogate each point cluster in the score plots. Several manually shaped masks may be tried until a satisfactory pixel class delineation is achieved for the features of interest in the multivariate image. The main ideas of this feature extraction technique of MIA have been illustrated through an application example on a 512 x 512 x 4 LANDSAT (MSS) multispectral satellite image. Pixels belonging to major roadways and paved areas were successfully segmented upon masking the corresponding point cluster in the score space.

The main ideas behind the feature extraction technique of MIA are further extended in the second half of chapter 3 to form the basis of a new on-line feature

monitoring scheme. This new scheme of monitoring features of interest from a sequence of time-varying multivariate images forms the main contribution of this thesis towards linking the fields of digital image analysis using multivariate statistical methods with industrial process monitoring. This technique employs the score space of the multivariate images to detect the number of pixels belonging to features of interest as the process progresses, with updated multivariate images being gathered intermittently.

However, an initial calibration step requires selecting a training multivariate image which is representative of all the features in the subsequent multivariate images that are of interest. Furthermore, if any known process faults exist, they must also be included in the training image if it is desired to monitor these faults as the process progresses. This training image is then decomposed into its score and image spaces using latent variable statistical methods to extract the features and known faults using feature extraction techniques of MIA, discussed in the first part of chapter 3. Once each feature and known faults of the process to be monitored have been associated to their respective score point cluster in the score space (using customized local area masks), the calibration of the training model is complete.

As the time-varying process progresses in real-time, the multispectral digital sensors transmit updated multivariate images at a predetermined sampling rate. Each multivariate image is rearranged into its two-way orientation and multiplied with the training model loading vectors to obtain the updated score space. Since the calibrated training model has already determined the spatial locations of the score point clusters corresponding to various features and process faults, the customized local area masks are used to simply count and highlight the number of feature pixels, or faults in the training image. This procedure is repeated for each updated multivariate image; thus, making it possible to monitor the progress of a time-varying process in real-time. A modified version of the LANDSAT (MSS) satellite image example is used in the second half of chapter 3 to illustrate a successful application of this process monitoring scheme. Several adjustments were made to the original multispectral satellite image to emulate a moving

satellite acquiring snapshots of the earth surface at a certain sampling rate while it moved over a fixed geographical region. Four features of interest were successfully monitored using this scheme through the 9 scans acquired by the moving satellite. Finally, chapter 3 discusses some computing time aspects to show the feasibility of the methods and their potential limitations. These timing results could also be used to compare applications of this methodology in other visually monitored time-varying processes.

This thesis represents a first attempt to link the fields of digital image analysis using latent variable statistical methods with time-varying process monitoring. No attempt has yet been made to develop any type of feedback control schemes pertaining to time-varying processes through MIA techniques. However, this thesis does build a platform upon which more work may be conducted in this area. As a result, this thesis opens the door for a lot of potential future research in developing further ideas and applications of MIA for the purposes of process monitoring, fault detection, and feedback control in fields where multispectral digital sensors are employed. The following subsection discusses some of the possible directions that may be pursued for future research projects in the field of latent variable statistical methods applied to multivariate digital imagery.

4.1 Directions for Future Research

The potential of applying latent variable statistical methods to multivariate digital imagery for the purposes of process monitoring and fault detection has been introduced in this thesis. However, this thesis should simply be considered a starting point from where further research may be pursued in the area of MIA applied to industrial processes that are equipped with automatic vision systems. This sub-section highlights five major directions that could be pursued for conducting some of the future research in this area.

4.1.1 Pattern Recognition & Image Classification using MIA

Since MIA is based on latent variable statistical methods this technique may also be employed to perform other off-line industrial process analyses. One such application of MIA is the off-line analysis of images for the purpose of classifying product quality. Research is currently being conducted to classify steel surface images based on the roughness of the surface texture. A large number of steel surface pits would indicate bad product quality. As a result, grayscale intensity images are acquired that represent several steel slab surface qualities. Pits are indicated by dark pixels, whereas bright pixels represent no pits. To analyze these images using MIA all intensity images are first converted to their respective frequency domains upon performing 2D-FFT, and representing the Fourier coefficient magnitudes as pixel intensities. The image representations of these 2D-FFT's are used to compress the information from the original images onto a common frequency base for classification using MIA. The main latent variable statistical methods proposed for this classification are Multi-way PCA, and Discriminant PLS. However, the results of this research are still incomplete and have been excluded from this thesis. Further research on this project will shortly be conducted to obtain satisfactory classification of the steel surface images. It should be pointed out that image classification using MIA on multispectral steel surface images has the potential of yielding better results as compared to the strategy described above.

4.1.2 Combining Traditional Image Processing Techniques with MIA

Traditional image processing techniques have been briefly reviewed in this thesis, with the aim of introducing the reader who has little knowledge of this field with some of its commonly used operations. As a result, further researchers in the area of MIA need not perform extensive reviews in the image processing literature. However, some of these operations may potentially be applied for preprocessing of images before applying MIA techniques. Thus, further research should investigate various applications of the

traditional image processing operations to complement the use of MIA techniques in on-line process monitoring, fault detection, feedback control etc. An example could involve the use of a certain filtering operation on a set of variable images from a multivariate image to ease the segmentation of a particular feature of interest prior to performing MIA operations.

4.1.3 MIA using Regression Model Based Latent Variable Methods

This thesis concentrated on the basic theory and applications of MIA using Multi-Way Principal Component Analysis (PCA) as the main latent variable statistical method to perform the analysis. Chapter 3 briefly listed several applications using other (regression based) latent variable methods (e.g. Principal Component Regression {PCR}, Partial Least Squares {PLS}) on multivariate digital imagery for off-line analyses. However, the full potential of using such methods for the purposes of on-line process monitoring, fault detection, and feedback control have not been researched. As a result, another direction for future work lies in analyzing the advantages of employing such regression model based statistical methods to monitor and detect features (and faults) of interest in real-time from sequences of multivariate images of time-varying processes.

4.1.4 Industrial Application of MIA On-Line Monitoring Scheme

A successful application of the MIA feature monitoring scheme was presented in this thesis using an example from the remote sensing field. However, these techniques have not yet been tested on real industrial time-varying processes equipped with multispectral digital sensors, transmitting sequences of multivariate image data. As a result, future work in this research project would include a real industrial application of these techniques for monitoring features of interest on-line. Only then would the power and shortcomings of this approach be realized, along with further clues that might help in increasing its efficiency.

4.1.5 Feedback Control Schemes using MIA

Finally, this research project is incomplete without making the eventual link between MIA and feedback process control. Future research might include an actual industrial application of information gathered using MIA for automatic feedback control of a time-varying process after performing various carefully planned designed experiments. These experiments would determine the process variables that could be manipulated to correct the problems giving rise to changes in the features of interest. Depending upon the type of process being monitored, an appropriate on-line feedback control scheme (on/off, global, or spatial) could be developed to automatically correct the detected faults as the process varies with time. As soon as the tolerance limits of the pixel densities in the score space are violated, the MIA monitoring system could either alarm the operator, shut down the process, or automatically send an appropriate corrective action signal to the manipulated variables, thus completing the feedback control loop.

REFERENCES

- Baxes, G. A., "Digital Image Processing: Principles and Applications," John Wiley & Sons, NY, NY, 1994.
- Berman, M., L.M. Bischof, E.J. Breen, and G. M. Penden, "Image Analysis," Materials Forum, 18, pp. 1-19, 1994.
- Ekstrom, M.P., "Digital Image Processing Techniques," Academic Press Inc., FL, 1984.
- Esbensen, K., and P. Geladi, "Strategy of Multivariate Image Analysis (MIA)," Chemometrics and Intelligent Laboratory Systems, 7, pp. 67-86, 1989.
- Esbensen, K.H., P. L. Geladi, and H. F. Grahn, "Strategies for Multivariate Image Regression," Chemometrics and Intelligent Laboratory Systems, 14, pp. 357-374, 1992.
- Geladi, P., and S. Wold, "Image Analysis and Chemical Information in Images," Analytica Chimica Acta, 191, pp. 473-480, 1986.
- Geladi, P., H. Isaksson, L. Lindqvist, S. Wold, and K. Esbensen, "Principal Component Analysis of Multivariate Images," Chemometrics and Intelligent Laboratory Systems, 5, pp. 209-220, 1989.

- Geladi, P., and K. Esbensen, "Can Image Analysis Provide Information Useful in Chemistry?" *Journal of Chemometrics*, 3, pp. 419-429, 1989.
- Geladi, P., and K. Esbensen, "Multivariate Image Analysis in Chemistry: An Overview, in J. Devillers and W. Karcher (eds.)," *Applied Multivariate Analysis in SAR and Environmental Studies*, Kluwer, Dordrecht, pp. 415-445, 1991.
- Geladi, P., and K. Esbensen, "Regression on Multivariate Images: Principal Component Regression for Modelling, Prediction and Visual Diagnostic Tools," *Journal of Chemometrics*, 5, pp. 97-111, 1991.
- Geladi, P., E. Bengtsson, K. Esbensen, and H. Grahn, "Image Analysis in Chemistry I. Properties of Images, Greylevel Operations, the Multivariate Image," *Trends in Analytical Chemistry*, vol. 11(1), pp. 41-53, 1992.
- Geladi, P., and H. Grahn, "Multivariate Image Analysis," John Wiley & Sons, Chichester, UK, 1996.
- Glasbey, C.A., and G.W. Horgan, "Image Analysis for the Biological Sciences," John Wiley & Sons, Chichester, 1995.
- Gonzalez, R.C, and P. Wintz, "Digital Image Processing," 2nd edition, Addison-Wesley Publishing Company Inc., Reading, MA, 1987.
- Grahn, H., N. M. Szeverenyi, M. W. Roggenbuck, F. Delaglio, and P. Geladi, "Data Analysis of Multivariate Magnetic Resonance Images I. A Principal Component Analysis Approach," *Chemometrics and Intelligent Laboratory Systems*, 5, pp. 311-322, 1989.

Grahn, H., N. M. Szeverenyi, M. W. Roggenbuck, and P. Geladi, "Tissue Discrimination in Magnetic Resonance Imaging: A Predictive Multivariate Approach," *Chemometrics and Intelligent Laboratory Systems*, 7, pp. 87-93, 1989.

Grahn, H. F., and J. Sä äf, "Multivariate Image Regression and Analysis, Useful Techniques for the Evaluation of Clinical Magnetic Resonance Images," *Chemometrics and Intelligent Laboratory Systems*, 14, pp. 391-396, 1992.

Hoskuldsson, A., "PLS Regression Methods," *Journal of Chemometrics*, 2, pp. 211-228, 1988.

Jackson, J. E., "A User's Guide to Principal Components," John Wiley & Sons, NY, NY, 1991.

Jähne, B., "Digital Image Processing: Concepts, Algorithms, and Scientific Applications," 2nd edition, Springer-Verlag, Berlin, 1993.

Jolliffe, I. T., "Principal Component Analysis," Springer-Verlag, NY, 1986.

Nomikos, P., "Statistical Process Control of Batch Processes," Ph.D. Thesis, McMaster University, Hamilton, ON, 1995.

Pratt, W.K., "Digital Image Processing," John Wiley & Sons, NY, NY, 1978.

Serra, J., "Image Analysis and Mathematical Morphology," Academic Press Inc. (London) Ltd., London, 1982.

Thompson, C.M., Shure, L., "Image Processing Toolbox for use with MATLAB®: User's Guide," The Math Works Inc., 1994.

Vilenkin, N. Y., "Stories About Sets," Academic Press Inc., NY, NY, 1968.

Wold, S., P. Geladi, K. Esbensen, and J. Ohman, "Multi-way Principal Components and PLS Analysis," *Journal of Chemometrics*, 1, pp. 41-56, 1987c.