

**Automated flaw detection method for
X-ray images in nondestructive evaluation**

by

Karl Wayne Ulmer

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Department: Electrical Engineering and Computer Engineering
Major: Electrical Engineering

Approved:

Signature  redacted for privacy

Iowa State University
Ames, Iowa

1992

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	v
CHAPTER 1. INTRODUCTION AND PROBLEM STATEMENT	1
CHAPTER 2. RADIOGRAPHY AND IMAGE PROCESSING OVERVIEW	5
2.1 Introduction	5
2.2 X-ray Imaging Process	5
2.3 Radiographic Image Characteristics	6
2.4 Image Acquisition	7
2.5 Review of Image Processing Techniques Useful to Radiography	9
2.5.1 General image enhancement	10
2.5.2 Noise filtering techniques	12
2.5.3 Feature extraction techniques	15
2.6 Future of Image Processing in Radiography	19
CHAPTER 3. A GENERAL AUTOMATED FLAW DETECTION SCHEME	21
3.1 Introduction	21
3.2 Attributes of an Automated Inspection System	21
3.3 Difficulties and Complications Faced in Developing an Automated Inspection Scheme	23
3.4 Review of Existing Automated Flaw Detection Schemes	24
3.4.1 Knowledge based scheme	24
3.4.2 Intensive processing based scheme	25
3.4.3 Discriminate feature based scheme	26

3.5 Development of an Initial Scheme	27
3.6 Description of Automated Flaw Detection Scheme	28
CHAPTER 4. AUTOMATIC NOISE REDUCTION	31
4.1 Introduction	31
4.2 Noise Reduction Problem	31
4.3 Successful Noise Reduction Techniques	33
4.4 Automated Identification and Measurement of Noise Characteristics . . .	40
4.5 Automated Noise Reduction Results	47
CHAPTER 5. AUTOMATIC TREND REDUCTION	49
5.1 Introduction	49
5.2 Trend Reduction Problem	49
5.3 Review Trend Reduction Techniques	50
5.4 Automatic Identification and Measurement of Trend Characteristics . . .	52
5.5 Automatic Trend Reduction Results	60
CHAPTER 6. GEOMETRICAL STRUCTURE REDUCTION	63
6.1 Introduction	63
6.2 Geometrical Structure Problem	63
6.3 Geometrical Structure Reduction Approach	67
6.4 Description of Geometrical Structure Reduction Technique	70
6.4.1 Identifying seed regions	71
6.4.2 Growing regions	72
6.4.3 Merging grown regions	75

6.4.4	Subtracting the estimate	76
6.5	Algorithm Description and Implementation	77
6.5.1	Identifying seed regions	78
6.5.2	Growing regions	81
6.5.3	Merging Grown regions	85
6.5.4	Subtracting the estimate	86
6.6	Geometrical Structure Reduction Results	86
CHAPTER 7. SPECIFIC PROCESSING STAGE		95
7.1	Quantitative Assessment of Processed Image	96
CHAPTER 8. SUMMARY AND CONCLUSIONS		98
8.1	Analysis of Processing Results	98
REFERENCES		103
APPENDIX A: NOISE FILTERING PROGRAM		108
APPENDIX B: MAIN DRIVER PROGRAM FOR VOSS TECHNIQUE		111
APPENDIX C: SURFACE TYPE CLASSIFICATION PROGRAM		115
APPENDIX D: INITIAL SEED REGION IDENTIFICATION PROGRAMS		118
APPENDIX E: REGION GROWING PROGRAMS		125
APPENDIX F: LEAST SQUARES FITTING PROGRAMS		132
APPENDIX G: SEED EXTRACTION PROGRAM		138
APPENDIX H: REGION MERGING PROGRAM		140

ACKNOWLEDGEMENTS

I wish to thank my advisor, Professor John P. Basart, for his guidance and encouragement throughout my graduate program. I wish thank the NSF Industry/University Center for NDE at Iowa State University for the funding they provided throughout my program.

I also wish to thank Joe Gray and Dick Wallingford for many helpful discussions concerning this work. I would also like to thank my committee members, Dr. Hsien-Sen Hung, Dr. Joe Gray, and Dr. Donald Thompson for their valuable time.

I am also very grateful for the steadfast encouragement and spiritual support of my wife, family and friends. I also wish to extend my deepest appreciation to my parents for their constant love and support.

LIST OF TABLES

Table 4.1:	Standard deviation of the noise and SNR's for test images in Figure 4.2	35
Table 4.2:	SNR and standard deviation of noise in filtered results for void type flaws	37
Table 4.3:	SNR and standard deviation of noise in filtered results for crack type flaw	39
Table 4.4:	Noise measures found for images in Figure 4.7	43
Table 4.5:	Noise measures found for images in Figure 4.9	46
Table 4.6:	SNR's for images shown in Figure 4.10	48
Table 6.1:	Surface type labels and values from H and K curvature signs	79

LIST OF FIGURES

Figure 3.1:	Flow diagram of proposed automated flaw detection scheme.	29
Figure 4.1:	Examples of digitized radiographic images possessing different noise characteristics: (a) weld area (Courtesy of Martin Marietta); (b) railroad frog (Courtesy of American Association of Railroads)	32
Figure 4.2:	Radiographic images used to evaluate noise filter performance: (a) crack type flaw; (b) void type flaws	34
Figure 4.3:	Noise filtering results for void type flaws	36
Figure 4.4:	Automated thresholding results for filtered images	37
Figure 4.5:	Noise filtering results for crack type flaw	38
Figure 4.6:	Automated thresholding results for filtered images	39
Figure 4.7:	Images used to illustrating ALV noise measure; clockwise from the upper left corner is the original, the adaptive smoothing filter result, 7x7 median filter result, and the result of the 3x3 MASF followed by a 3x3 median	42
Figure 4.8:	Slices of image data illustrating noise and signal suppression for different noise filtering techniques; plot shows same row position; the top slice is from the original image, the remaining slices are from the ASF, MASF, 3x3 median of MASF, and 7x7 median filtering results, respectively	44
Figure 4.9:	Images used to illustrating ALV noise measure; clockwise from the upper left corner is the original, the adaptive smoothing filter result, 7x7 median filter result, and the 3x3 MASF result	45

Figure 4.10:	Nonuniform images containing a large range of noise levels which are listed in Table 4.6	48
Figure 5.1:	Examples of radiographic images containing background trends: Courtesy of Atlantic Richfield, Westinghouse, and American Association of Railroads	51
Figure 5.2:	Radiographic image of a fuel tank weld (Courtesy of Martin Marietta): (a) before; (b) after trend removal	54
Figure 5.3:	Surface plot of intensities in Mart	55
Figure 5.4:	Reconstructed plot of the least squares lines calculated for the selected rows (a) and columns (b)	56
Figure 5.5:	Intermediate surface plots of Mart demonstrating the zero order fitting technique: (a) half done with column fitting; (b) completed column fitting	57
Figure 5.6:	Intermediate surface plots of Mart demonstrating the zero order fitting technique: (a) half done with row fitting; (b) completed row fitting .	58
Figure 5.7:	Flow diagram of automatic trend reduction subprocess	59
Figure 5.8:	Radiographic images before and after trend removal: (a) originals; (b) results after trend removal; top image is from the fuel tank weld referred to previously, and bottom image is a composite material (Courtesy of Westinghouse)	61
Figure 5.9:	Radiographic image of pipe weld before and after trend removal: top left image is original; image to the right is histogram equalized version of original; bottom left is approximated background trend; bottom right is result after trend removal	62

Figure 6.1:	Examples of radiographic images containing geometrical background structures	65
Figure 6.2:	Results from existing techniques used to reduce variations due to geometrical structures	68
Figure 6.3:	Flow diagram of four main steps in the approach to reduce the effects of geometrical structure	69
Figure 6.4:	Simulated radiographic image of air conditioner part showing region of interest	71
Figure 6.5:	Results from classifying pixels into fundamental surface types: (a) surface plot of intensities in region of interest; (b) pixel classification results	72
Figure 6.6:	Image showing the four main seed regions identified.	73
Figure 6.7:	Sample region shapes and corresponding topological descriptor values .	74
Figure 6.8:	Surface plot of intensities in grown region: (a) least squares approximation of actual image surface in (b)	75
Figure 6.9:	Comparison of original (a) and reconstructed images (b)	76
Figure 6.10:	Geometrical reduction results: (a) residual image; (b) noise processed image	77
Figure 6.11:	Eight fundamental surface types used to classify each pixel	80
Figure 6.12:	Flow diagram showing decision rules for accepting newly grown region	84
Figure 6.13:	Geometrical reduction results: (a) original image; (b) residual image .	87
Figure 6.14:	Plot of row data in original image	89
Figure 6.15:	Plot of row data in residual image	89

Figure 6.16:	Thresholding results for image in Figure 6.13b	90
Figure 6.17:	Portion of original radiographic image of air conditioner part	91
Figure 6.18:	Approximated image intensities found using VOSS technique	92
Figure 6.19:	Residual image intensities found by subtracting approximated image from original	93
Figure 6.20:	Candidate flaw locations found by thresholding the residual image in Figure 6.19	94

CHAPTER 1. INTRODUCTION AND PROBLEM STATEMENT

Private, government and commercial sectors of the manufacturing world are plagued with imperfect materials, defective components, and aging assemblies that continuously infiltrate the products and services provided to the public. Increasing awareness of public safety and economic stability has caused the manufacturing world to search deeper for a solution to identify these mechanical weaknesses and thereby reduce their impact.

The areas of digital image and signal processing have benefited greatly from the technological advances in computer hardware and software capabilities and the development of new processing methods resulting from extensive research in information theory, artificial intelligence, pattern recognition and related fields. These new processing methodologies and capabilities are laying a foundation of knowledge that empowers the industrial and academic community to boldly address this problem and begin designing and building better products and systems for tomorrow.

The Nondestructive Evaluation (NDE) community has been actively involved in utilizing these developments to improve existing inspection techniques that measure a material's characteristics to detect, size, and classify defects in the material without destroying it. NDE is becoming increasingly important as competition forces businesses to improve the efficiency of the manufacturing process by identifying and removing defective materials and components from the process before valuable resources such as time, labor and material have been invested. NDE is also a valuable tool for inspecting

the various components of the end-product throughout its lifetime. This is especially true when public safety is dependent on the product's ability to operate reliably.

Several NDE inspection techniques are commonly used in industry today. These include ultrasonic, eddy current, and X-ray inspection techniques. Less commonly used techniques include acoustic emissions, fluorescent dye penetrants, holography, and shearography. Some of these techniques are limited by their performance and feasibility outside of the laboratory setting.

X-ray inspection techniques perform effectively in a variety of NDE applications and can benefit significantly from the advances in image and signal processing. In the X-ray imaging process, a specimen is illuminated by an electromagnetic field that propagates through the specimen to project a 2-D image on the film plane lying below. This projection represents the spatial density distribution of the material in the direction of the ray. Higher material densities or thicker material attenuate the ray and limit the number of X-ray quanta reaching the film plane. This yields a brighter film density. Conversely, lower material densities or thinner material yield darker film intensities. Hence, material with a crack or void type flaw yields a darker film area, associated with the flaw, superimposed on a lighter background associated with the material surrounding the flaw.

Radiographs are inspected by persons who are thoroughly trained to recognize these variations in film density due to flaws in the presence of nonuniform background and other unwanted image features. The operator's success in detecting the flaw signals is limited by the sensitivity of his/her eyes. Additional factors affecting successful inspection include eye strain and operator fatigue which result in poor detection performance. These and other factors contribute to the fluctuating quality of inspection achieved in

the conventional radiographic inspection process.

Image processing techniques possess the ability to enhance the visual quality of the images for human inspection and interpretation. Image restoration, feature extraction, pattern recognition, and scene interpretation techniques have been successfully applied to a variety of different imaging applications. These applications include radio astronomy, remote sensing, machine vision, medical diagnostics, commercial photography, and X-ray applications.

Image processing techniques are able to aid the operator in successfully inspecting the radiograph by enhancing the flaw signal in the presence of high noise, low contrast, and nonuniform background. Nonetheless, manual radiographic inspection is still a time consuming process. The utilization of NDE radiography in an increasing number applications and the volume of radiographs requiring visual inspection in each particular application place a great strain on the conventional inspection process.

The objective in this research was to develop version 1 of a general automated radiographic inspection scheme capable of detecting, measuring, and classify flaws with minimal human interaction. This was achieved through the utilization of existing image interpretation and measurement techniques and the development of new techniques that are presented herein.

This thesis presents some background information in Chapter 2 on radiography and film characteristics and discusses digital image acquisition and processing tools found useful in that area. Chapter 3 discusses the requirements of a general automated flaw detection scheme, considers the problems faced in developing such a scheme, surveys existing schemes, and presents the proposed scheme. The three chapters following present the automated processing techniques contained in the scheme. Specifically, Chapter 4

focuses on automated noise reduction processes, Chapter 5 presents automated trend reduction techniques, and Chapter 6 presents techniques for automatically reducing the effects of the specimen's nonuniform geometry. Chapter 7 then comments on specialized processing methods for automatically extracting and classifying candidate flaws. A summary of the automatic flaw detection scheme is presented in Chapter 8 which also suggests areas of further investigation.

CHAPTER 2. RADIOGRAPHY AND IMAGE PROCESSING OVERVIEW

2.1 Introduction

The first section of this Chapter presents a brief discussion of the X-ray imaging process and the image characteristics associated with radiography. This is followed by an overview of image acquisition process and system requirements. The later portion of the Chapter is dedicated to reviewing both general image processing techniques and techniques with special application to the radiographic image processing field.

2.2 X-ray Imaging Process

The X-ray image formation processes [15] is attributed to both controlled and uncontrolled physical processes and conditions. The X-ray image formation process is a complicated process based on the attenuation of an electromagnetic field propagating through an object. The strength of the electromagnetic field and the time of exposure is controlled to yield a radiographic image with optimal contrast sensitivity and noise characteristics. A radiographic image is formed by measuring the number of photons passing through the material at a particular location on the film plane. The measurement device can be an X-ray film that is sensitive to photons or an image intensifier tube that converts the invisible photons to visible light. X-ray film is most commonly used in industrial radiography today.

Some of the industrial applications encountered include inspection of welds, railroad rails, valves, turbine blades, and other crucial components. The ability to control the strength of the electromagnetic field enables one to utilize X-ray techniques in a variety of industrial inspection applications independent of the diversity in material absorption or attenuation characteristics. However, uniform signal sensitivity throughout the image suffers when extremes in material absorption are present in a single specimen.

2.3 Radiographic Image Characteristics

The image content produced by the X-ray image formation process includes image features such as flaw signal, noise, background trends, and intricate image background structure due to complex specimens. The radiograph is viewed as an image of a flaw corrupted by these three unwanted image features; noise, background trends, and intricate image background structure. To achieve accurate inspection results, successful interpretation of the image is crucial. The origin of each of the unwanted image features is discussed below.

Noise within the film is one of the uncontrolled processes in the X-ray image formation process and is due to the quantum nature of the electromagnetic radiation. Additional noise is signal dependent according to the Poisson statistics model of the image formation process. Radiographic noise characteristics are further discussed in Section 4.2.

Background trends are variations in film density due to nonuniform variations in material density or thickness. These slow global changes result from continuous variations in material density or thickness that span across the radiographed specimen. The

characteristics of these trends, background trend examples, and the problems associated with them are presented in Section 5.2.

Background structure or geometric structure of the material can cause local variations in film density. Unlike background trends with global characteristics, these variations can be attributed to local changes in material thickness or density that are associated with the shape of the radiographed specimen. The general characteristics of the variations due to geometrical structures, radiographic image examples, and the problems associated with them are presented in Section 6.2.

2.4 Image Acquisition

The process of transforming the X-ray films into digital images which can be interpreted by computers is known as image acquisition. The quality of the equipment used in this process has a significant role in controlling the quality of the digitized image. An image acquisition system consists of three main devices, a light box, a video camera, and a frame grabber board. The light box is used to illuminate the film which is then imaged by the television camera and captured by the frame grabber board. Each of these devices plays an important part of the image acquisition process.

A good quality light box will possess uniform lighting characteristics across its surface and have a variable brightness control [9]. The range of brightness should enable viewing of film densities ranging from 1.0 to 4.5. Different radiographs will emphasize different film density ranges and require different light box brightness levels to enhance the film information. The video camera used to image the illuminated film requires flexibility in obtaining the optimal image settings. The camera lens should possess zooming capabilities and several aperture settings to adjust to the needs of the

radiograph. The camera stand should also be capable of vary the camera to film distance. Other adjustable settings found on the camera control unit may include white balance and black level control. The COHU 5300 Camera Control unit used in the image processing laboratory in the Electrical Engineering and Computer Engineering department at Iowa State University has automatic white balance and black level control capabilities. The automatic control may be helpful in an industrial setting, but manual control was found to be more reliable in obtaining the optimal image settings for each particular radiograph.

The imaging process converts the spatially varying film intensity into a 2-D array of numerical values that can be processed by a computer. The size of the 2-dimensional array is determined by the size of the image device. The dimensions of the CCD array in the camera used at Iowa State University's Image Processing Laboratory is 512 units wide by 480 units long. These dimensions relate to the number of discrete spatial samples used to quantize the width and length of the X-ray film image being captured.

Each discrete sample location on the film is represented by a corresponding location on the imaging array called a pixel. The amount of light from each sample location on the illuminated film is quantized into a finite intensity value called a grey level. This finite value is assigned to the corresponding pixel location in the image array. The number of possible grey levels is limited by the resolution of the device. Hence, an 8-bit image device has 256 possible grey levels and a 12-bit device has 4096. The higher resolution devices are more sensitive to variations in contrast and therefore yield a more accurate representation of the original radiograph. However, higher resolution devices are also more sensitive to noise and require greater storage capacity.

An array of values representing an image is called a frame. It takes approximately 1/30 of a second for the frame grabber board to acquire a frame from the television

camera. The light quantization process introduces quantization noise into the image by mapping the observed continuous light into discrete grey levels. Frame grabber boards can reduce the effects of this noise by averaging together several frames to obtain a more accurate approximation of the light levels transmitted from the original radiograph. When averaging N frames, the RMS noise value is proportional to $1/\sqrt{N}$ when each sample frame is statistically independent. The acquired image can then be viewed by converting the pixel values into light intensity and displaying the digitized image array on a video screen.

When the proper settings are made on the digitizing system, the resulting digital image can be enhanced and interpreted by intelligent image processing algorithms that aid in the accurate identification and classification of features in industrial radiographic images.

2.5 Review of Image Processing Techniques Useful to Radiography

The development of image processing routines for industrial radiographic applications is a time consuming and complicated venture. Although many image processing techniques can be adopted from other application areas such as medicine, astronomy, and remote sensing, the nature of the radiograph and features within it require some special enhancement functions not common in other areas. Some image enhancement processes found to be applicable to a variety of application areas including radiographic images have been implemented by the image processing group in the Electrical Engineering and Computer Engineering department at Iowa State University. The following will be a review of both general and specific image enhancement processes. These processes have been divided into three categories, general image enhancement, noise filtering, and feature extraction techniques.

2.5.1 General image enhancement

Many enhancement techniques involve simple manipulation of the intensities within an image. These techniques are based on either global pixel characteristics or relationship among pixels in local neighborhoods. In many cases these techniques are performed on the image data in both the spatial and frequency domains. The goal of all these general enhancement techniques is to enhance the image's appearance without introducing artifacts.

Some of the simplest forms of image enhancement are performed using histogram modification techniques [19, 34]. These techniques take advantage of the information present in the histogram, a function describing the distribution of intensity levels in an image. When an image is mostly dark or mostly bright, the distribution of the intensity levels is somewhat concentrated and the contrast in these images is low. In these situations, histogram modification techniques are very effective in improving the images appearance.

Histogram modification techniques include contrast stretching and histogram equalization. Contrast stretching redistributes the recorded image intensities throughout the range of possible grey levels. Histogram equalization is a technique used to obtain a uniform histogram. This process spreads the recorded values throughout the entire range of grey levels in such a way that every intensity level occurs with equal probability. This has the effect of increasing the contrast between consecutive intensity values occurring with higher probability. Histogram modification techniques are useful in enhancing details not so evident to the human eye.

Other methods proposed to enhance the image contrast have been based on local contrast information. Recently, a new contrast enhancement approach was introduced

by Khellaf et al. [22]. This method was derived from entropy concepts in information theory and uses local contrast information to determine the entropy for each individual pixel. The method incorporates both median filter and histogram equalization techniques to obtain superior results over other histogram-based methods.

Image contrast is closely related to the gray-level gradient. Large gray-level gradients correspond to large changes in image information. Therefore, local contrast can be enhanced by amplifying these local changes. Discrete methods used to measure gradient information are commonly referred to as edge detection techniques since they emphasize the large gradients occurring at edges. Common edge detection operators include the Roberts, Prewitt, and Sobel operators [14]. The gradient information in digitized images is approximated by differences. The magnitude of the gradient at $f(x,y)$ in a digital image can be found using the relation

$$\| f_{xy}(x, y) \| \approx \left\{ [f(x, y) - f(x + 1, y)]^2 + [f(x, y) - f(x, y + 1)]^2 \right\}^{1/2}. \quad (2.1)$$

A better approximation of the magnitude of the gradient can be found by including more distant neighbors in the approximation. The calculation of the gradient over larger regions also reduces the effects of noise. However, this also tends to blur the edge positions and requires a significant amount of additional computations.

The magnitude of the gradient reflects the sharp variations due to edges independent of their direction. This results in identifying edges with high gray-level values and smooth regions with low gray-level values. Other useful edge detection methods are based on image entropy [38] and mathematical morphology [36].

In general, the above enhancement techniques do not discriminate against noise, and therefore amplify the noise as well as other image features. One form of image enhancement involves reducing these spurious effects by smoothing the image. Neighborhood averaging is one such common smoothing technique. It is performed in the spatial domain and works on the basis that neighboring pixels have similar intensities. The averaging process tends to blur out small distracting image features and enhances the global or larger scale features. Another effective smoothing technique is low pass filtering. This technique is performed on images in the frequency domain and can be used to attenuate high frequency content in the image commonly attributed to noise and edges.

High pass filtering works on the same principle except that it is used to attenuate the low frequency content of the image. This results in enhancing edges and other abrupt changes in intensity that are associated with high frequency components.

Unsharp masking is yet another general enhancement technique applied to emphasize image details. Unsharp masking is achieved by first applying neighborhood averaging and then subtracting the resulting image from the original. The residual image reflects the sharp transitions or variations not easily distinguished in the original image. This technique often accentuates the image's noise content making it a valuable noise characterization tool.

Most of the general image enhancement techniques described above require few if any input parameters. These nonparametric techniques can be easily performed on a great variety of images to aid in the interpretation process.

2.5.2 Noise filtering techniques

Many enhancement techniques are focused on image restoration in terms of reducing noise and enhancing true image signals. The image restoration problem involves esti-

inating the original image from a distorted image based on information about the type of noise and the image formation process in terms of the point spread function (PSF). Because there is no unique restoration solution, there are a tremendous number of journal articles concerning noise filtering techniques and hence a great number of approaches to the problem.

Many of the noise filters are based on mathematical model given by

$$g(i, j) = s\{H[f(i, j)]\}^+ n(i, j) \quad (2.2)$$

where $g(i, j)$ is the recorded image, s is the imaging system response, H represents the image formation process, $f(i, j)$ is the original image, and $n(i, j)$ is the noise process that may be additive, multiplicative, or a combination of the two.

Basic assumptions about the image formation process and type of noise present in an image allow one to tailor a noise filtering technique to images with those particular characteristics. Examples of such techniques include inverse, Wiener, power spectrum equalization, and Kalman filters [25, 47, 46]. Zheng developed an adaptive Kalman filter to distinguish between two or more additive stochastic processes whose spectra overlap. In this filter, the parameters are adapted to the local features in each segmented region of an image.

Some techniques are not based on an image model. These include median filters which are nonlinear and like most spatial filters are applied using windowing techniques. Windowing techniques assume a special relationship among neighboring pixels and therefore limit the filtering operation to local neighborhoods. Windowing is performed by centering a $(2n+1) \times (2n+1)$ window on the pixel that will receive the result of the filtering operation. Each pixel in the image is processed independently in the windowing

technique. Because of this independence, filters based on windowing techniques are readily adaptable to real-time image processing applications.

In median filtering, the operation involves selecting the median value from the pixel values lying in the 2-D window and assigning this value to the center pixel. Many variations of this nonlinear filter exist. The median filter is useful in radiographic image processing and many other applications due to its ability to smooth out spurious noise while preserving edge boundaries.

Other nonlinear methods that smooth an image while preserving discontinuities are based on iterative weighted averaging methods [26, 44, 35]. Unlike the others, the method proposed by Saint-Marc et al. [35] guarantees convergence. This technique is performed by iteratively convolving the image with a small weighted averaging mask whose weights reflect, at each point, the continuity of the local region. At convergence, the image will consist of constant intensity regions separated by step edges. Prior to convergence the process sharpens the edges and smooths regions surrounding the edges. This technique was an integral part of the modified adaptive smoothing filter mentioned in Section 4.3.

Other adaptive noise filtering techniques are model based. Lee developed some of the first successful model based techniques using local statistics [26]. These techniques are robust in that they adapted to applications where the signal mean and noise variance are spatially varying and unknown. The local statistics used are local mean and local variance. The algorithms are based on the assumption that the sample mean and variance of a pixel are equal to its local mean and variance which are calculated from pixels lying in a local window. Continued use of these assumptions by others [27, 24] to developing new noise filtering algorithms shows the significance of local statistics. These techniques display the ability to smooth out the noise while preserving edges within the image. The

basic form used by Lee and others is shown in Eq. (2.3), where $\hat{f}(i,j)$ is the filtered image, $f(i,j)$ is the original image, $g(i,j)$ is the degraded image, $v_f(i,j)$ is the local spatial variance of $f(i,j)$, and σ_n^2 is the noise variance found from a priori measurements. Lee refers to this as a nonstationary mean nonstationary variance (NMNV) image model.

$$\hat{f}(i,j) = \bar{f}(i,j) + \frac{v_f(i,j)}{v_f(i,j) + \sigma_n^2(i,j)}(g(i,j) - \bar{g}(i,j)) \quad (2.3)$$

The performance of the noise filter depends greatly on the accuracy of the methods used to calculate the local spatial means of $f(i,j)$ and $g(i,j)$, which are represented by $\bar{f}(i,j)$ and $\bar{g}(i,j)$ and the local spatial variance $v_f(i,j)$ from the degraded image $g(i,j)$. Various methods of computation perform differently based on the image characteristics. Noise filtering techniques based on these local statistics have been developed for both additive and multiplicative noise [24]. In their paper, Kuan et al. [24] develop a similar noise filtering model for images with Poisson noise. This filter is especially useful in radiographic image processing applications.

2.5.3 Feature extraction techniques

Thus far, the discussion of image processing techniques has not been limited to any specific application. The previously mentioned techniques are applicable to most all areas of image processing. However, feature extraction implies that the information of interests within the image contains certain attributes associated with the application at hand. Therefore, the feature extraction techniques discussed here are based on a prior knowledge concerning the characteristics of the image features to be extracted in radiographic images.

Feature extraction techniques can be used to enhance a particular feature within an image by two methods. One method is to perform an operation that will enhance the desired image feature, feature enhancement. The other method involves removing unwanted image features, feature suppression. As mentioned before, there are four main features present in radiographic images. These features include flaws, noise, background trends, and geometrical structure. The approach in this thesis is to use feature extraction techniques to enhance the flaw signal and suppress the other unwanted image features within the radiographic image.

In nondestructive evaluation using radiography, the desired image feature is the flaw. Flaw enhancement can be performed by several different techniques. Each technique may vary depending on the amount and type of a prior information necessary to successfully enhance the flaw. Correlation techniques [14], also known as matched filter techniques [43], are used to determine if an image contains a region similar to the one described by a 2-D mask. The mask incorporates prior information concerning physical flaw characteristics such as shape, size, and orientation of the flaw. When these three parameters are properly defined, the matched filter is a very successful feature extraction technique.

Mathematical morphology has also been found effective in extracting flaws. The technique is called morphological filtering [6, 41] and, like matched filtering, it is performed using a mask to describe the shape, size, and orientation of the flaw. Morphological filtering requires additional information concerning the relationship between the intensity of the flaw and its surroundings. Morphological techniques are effective when the images are not very noisy.

Techniques used to reduce unwanted image features, and thereby enhance the desired

feature, have also been successful in radiographic image processing applications. Such specialized techniques include trend removal [11] and background subtraction [6]. Trend removal and background subtraction remove the global image intensity variations caused by continuous changes in material density or thickness within the specimen. These intensity variations inhibit automatic flaw detection algorithms from producing accurate results. Trend removal is performed by fitting a polynomial model to the image by least-squares, and then subtracting the approximated model from the original image to obtain a residual image. The polynomial model is based on the expected trend characteristics. The polynomial model can be either one or two dimensional and may possess up to third order polynomial terms. Variations of the polynomial trend removal technique are thoroughly discussed by Doering [11]. Background subtraction is based on the idea that the small flaws will be suppressed by a large neighborhood averaging filter. The residual image found by subtracting the smoothed image from the original is found void of large scale background features and better prepared for automatic flaw detection. In addition to smoothing techniques, background estimates can also be found using morphological techniques. Trend removal and background subtraction techniques are only effective on images void of intensity variations due to geometric structure.

Image segmentation is another feature extraction method found useful in many image processing applications including radiography. Image segmentation is a process that organizes image pixels into connected regions such that each region contains pixels with similar properties. The earliest form of image segmentation involved grouping pixels in the histogram with approximately equal intensities. This histogram-based form of image segmentation is commonly referred to as thresholding [43]. Different approaches for selecting the optimal thresholding value include the identification of a global thresholding

value [28, 31] and a spatially varying threshold value found using either contrast measures [37, 29] or iterative techniques [33]. The similarity measure used to group pixels in these algorithms is equivalent gray level. Therefore, histogram-based techniques are only useful when a range of intensity levels is unique to a region representing a desirable image feature. This is most often not the case in radiographic images corrupted by trends and geometric structure. However, these techniques have been found to be quite useful in a multitude of applications including the final stages of flaw detection.

Image segmentation based on region growing techniques [14, 45, 18, 43] has been studied extensively in the image processing and pattern recognition community. Region growing techniques are useful in partitioning an image into meaningful regions. The main idea behind region growing techniques is simple. After starting points are identified within an image, adjacent pixels are examined one by one and tested to see if they possess properties similar to those of the current region. The pixels are accepted into the region if their properties are similar or rejected if their properties are inconsistent. Testing for similar properties is achieved via a uniformity predicate $P()$. A common definition of segmentation by Horowitz and Pavlidis [18] states that if I is the set of all image pixels and $P()$ is a *uniformity predicate* defined on groups of connected pixels, a segmentation of I is a partitioning set of image regions $\{R_1, \dots, R_N\}$ such that

$$\bigcup_{i=1}^N R_i = I \quad \text{where} \quad R_i \cap R_j = \emptyset \quad \forall i \neq j, \quad (2.4)$$

the uniformity predicate $P(R_i) = \text{TRUE}$ for all regions, and

$$P(R_i \cup R_j) = \text{False} \quad (2.5)$$

whenever R_i is adjacent to R_j . The uniformity predicate is simply a test that is performed on a group of connected pixels to determine if they possess similar image characteristics.

Hence, image segmentation applications based on different image characteristics are developed using different uniformity predicates. These uniformity predicates are determined by knowledge about the type of scene being viewed and the image formation process.

The simplest form of uniformity predicate would be equivalent intensity level. A region growing procedure based on intensity level alone [32] would be comparable to a histogram-based technique. A uniformity predicate found successful in classifying terrain in Landsat images is texture [4, 39, 7]. A uniformity predicate developed for range imaging applications is based on the idea of approximating the pixel data in a region by a surface function [16, 13, 30]. The approximated surface functions are used to identify relevant structure within the range images. In the same way, these surface functions can be used to identify the intensity variations in a radiographic image resulting from geometrical structure. After approximating the image intensity variations due to geometrical structures via region growing techniques, feature suppression techniques can be applied to reduce the effects of the unwanted structures and enhance the flaw signals.

Image segmentation techniques based on region growing and surface estimation hold much promise in the goal to develop automatic procedures capable of identifying and reducing the effects of geometrical structure in radiographic images.

2.6 Future of Image Processing in Radiography

Within the area of industrial radiography there are an enormous number of different applications that each produce a variety of radiographic images requiring manual inspection. Some of the main features common to these radiographic images have been identified and commented on above. In addition, image processing techniques have demonstrated the ability to enhance radiographic images for improved inspection perfor-

mance. This has been achieved through the development of image processing techniques capable of reducing the unwanted features and enhancing the desirable features. Continued research and development in the area of image interpretation algorithms will pave the way for automated radiographic inspection systems.

CHAPTER 3. A GENERAL AUTOMATED FLAW DETECTION SCHEME

3.1 Introduction

This Chapter begins with a discussion on the requirements of an automated flaw detection scheme and the difficulties and complications encountered in the planning process. It then reviews previous approaches to the automated inspection problem. The following sections present important items considered in the development of the initial scheme and describes the scheme itself. The last section comments on the importance of each individual part of the scheme and states the main contribution of this thesis.

3.2 Attributes of an Automated Inspection System

I will illustrate the problem faced in developing an automated inspection system. In the radiographic inspection process, the operator is required to analyze a particular set of radiographs and identify candidate flaws. Imagine for a moment that the operator is not aware of the type of geometry or structure associated with the radiographed area of the specimen. Nor is the operator familiar with the appearance of the types of flaws most likely encountered. Without this knowledge, the likelihood that the operator will successfully interpret just one radiograph is not favorable. Under these circumstances, the inspection process would be sufficiently more difficult and the operator would most likely misinterpret variations in material structure and overlook candidate flaws. This is the situation presented to the computer in an automated flaw detection problem. In fact,

the situation looks even worse for the computer when you consider that the operator has the added capability of interpreting a scene through the astoundingly powerful human vision system.

An automated inspection system requires intelligent image interpretation software. The system will execute the commands programmed into the computer by the software developer. The system's interpretation of the features in the radiographic image are based wholly on the image measurements that the developer has considered relevant. The system is developed in the hope that all significant image information has been acquired to enable the system to make an accurate interpretation of the radiograph. This type of system is commonly referred to as an 'expert system'.

An expert system designed to detect flaws in radiographic images would perform the same basic steps that are performed in the manual inspection process. The manual inspection process involves more than a relatively quick glance at the radiograph. It involves preliminary examination of the global features within the image. This review of the image provides the inspector with a general idea of the type of background trends and geometrical structure present in the image. From this analysis, the inspector is better able to discriminate between intensity variations produced by flaws and those due to background trends or geometrical structure.

In the same way, an automated inspection system would also be required to accurately discriminate among the various features found in an arbitrary radiographic image and process the image in such a way that a flaw visually present within the image would not go undetected. In addition to detecting all possible flaws, the system should also achieve low false alarm ratings. Furthermore, the system should perform the inspection in as little time as possible.

3.3 Difficulties and Complications Faced in Developing an Automated Inspection Scheme

The previous section states the requirements of an automated inspection system. To achieve this, the inspection scheme must be flexible in providing for the incorporation of prior information concerning the image or flaw characteristics. The scheme must also be designed in a manner that provides for intelligent decision making procedures that automatically decides which processes are necessary and which techniques would be most effective and efficient. These requirements present a number of problems in developing such a system.

Some of the most difficult tasks are faced when trying to determine which image processing functions are necessary and which particular method would be most effective and efficient in performing that function. Formidable tasks are encountered in the development of the automated decision making procedures and the creation of relevant image measurements that provide the necessary image information for selecting the appropriate processing techniques. It is difficult to know how to go about inventing mathematical measurements that can quantitatively describe a physical image characteristic. Many times a measurement is not consistent in fully describing a particular image characteristic. Other times the measurement may not be able to differentiate between the desired image characteristic and another similar, but unimportant, characteristic.

This process is further complicated by the need for accurate and reliable image processing techniques capable of performing the desired function. The field of image processing has come a long way in developing new techniques, but many image interpretation problems are far from being solved. Problems encountered in the techniques used in automated X-ray inspection include both sensitivity of feature extraction methods to

noise and insufficient discrimination between local flaw features and regular geometric structures.

These are some of the complications encountered in developing adequate measurements, reliable decision making procedures, and useful image interpretation techniques.

3.4 Review of Existing Automated Flaw Detection Schemes

Automatic inspection of industrial parts with X-rays has been pursued by many people in recent years. In addition, the machine vision community has also become strong in the area of automated industrial inspection and monitoring systems. However, both of these areas are plagued by application specific systems that require significant modification when adapted for other applications. This is also the case for the automated radiographic inspection systems reviewed here.

3.4.1 Knowledge based scheme

One of the greatest demands for automated radiographic inspection is in the area of weld defect identification. A knowledge based system was developed in Japan [23] for this particular application. The system is based on the interactive adjusting of 40 different parameters that are used to control everything from extracting the weld boundary to selecting the appropriate threshold for feature extraction. After an exhaustive process of iteratively adjusting the parameters and monitoring the systems performance, a fixed set of parameters is saved to disk and used to automatically process a given set of radiographic weld images.

The method begins by extracting the weld area and filtering it with two modified versions of an unsharp masking filter. After thresholding the processed images, some of the parameters are used to eliminate smaller candidate flaws based on their size and shape.

Limited results showed that the system could properly classify the defects after they had been identified using the proper parameter settings. This processing approach suggests the use of simple techniques for extracting the weld boundary and simple filtering techniques for identify the candidate flaw locations. The controlling factors of this processing approach come into play in finding the optimal settings for the 40 different parameters used in the system. These settings are dependent on the application at hand and may vary significantly for different X-ray exposure settings or material characteristics.

3.4.2 Intensive processing based scheme

A similar software package was developed for Martin Marietta Corporation by the Image Processing Group in the Electrical Engineering and Computer Engineering department at Iowa State University. This system was designed to read in an image, process the image to identify candidate flaw locations, classify and assign a confidence level to candidate flaws, and then record the results in a data base. The processing portion of the program was divided into three main steps. First, like above, the weld boundary was identified. Extensive work was performed in this area to developing techniques to accurately identify the weld boundary that could easily vary depending upon the width of the weld bead. Next, the portion of the image within the weld area was divided into three overlapping sections that would then be exposed to a series of filters designed to extract the defects lying in each particular section. These filters were designed to extract the flaws without enhancing features due to noise or other phenomena, within the weld area, such as intensity variations due to the changing material thickness along the weld bead edge. The third step involved thresholding the filtered images to identify candidate flaw locations. Sometimes, additional processing was performed on the immediate area surrounding a candidate flaw to better determine a flaws presents. The thresholded flaw characteristics

such as size, location, and shape were then calculated to help in determining a confidence level for each flaw. This information was then stored in a database for later reference.

This software package was developed on a SUN SPARCstation using a MC68030 based microprocessor. The time involved for processing a single image was approximately 10 minutes. The processing time could be significantly reduced by implementing the scheme on a parallel processing machine. The processing portion of the program had adjustable parameters, but they were not made easily accessible by the end user and thus the system was found to be unsuccessful in accurately identifying flaws while minimizing the number of false flaw identifications. Knowledge acquired from this package is being used by Martin Marietta to develop a faster and more reliable inspection system on a parallel processing platform.

3.4.3 Discriminate feature based scheme

Another example of an automated radiographic inspection system is found in Boerner [3]. Like the others, this system was designed with a single application in mind. This system automatically inspects aluminum wheels for flaws by rotating them around their axis and inspecting them at three different test positions: one for the hub, one for the spokes, and one for the road wheel. The system was designed with some features that could be very useful in other inspection applications. One of these features was that of calibrating the image so that there resulted a linear relationship between the gray value in the image and the thickness of the material being analyzed. This proved beneficial in sizing candidate flaws.

Another feature of interests was discussed in the identification or extraction of flaws. Their method involved the combined use of orthogonal and rotationally invariant discriminate features to identify the flaws. The discriminate features were identified

using one training set of expected flaw signals and another set including the remaining signals due to unwanted features in a single test image of the hub. After calculating the features based on the training sets, the features were used to successfully classify each pixel in a different hub image into one of two classes: pixels possessing flaw-like characteristics and those without. In addition to properly identifying flaws in a similar image, the features were also used to successfully identify flaws in the more geometrically complicated spoke image. This demonstrates the power of orthogonal and rotationally invariant feature discrimination techniques. However, it should be noted that these results could not be obtained with image corrupted by a significant amount of noise.

This review of various approaches to an automated X-ray inspection system reveal some of the difficulties and complications discussed above in developing such a system.

3.5 Development of an Initial Scheme

In the process of developing a valid general automated flaw detection scheme, there are many factors to consider. One of the main factors is that of being applicable to a wide variety of radiographic applications. This would require the scheme to be robust in dealing with radiographs containing a literally innumerable kinds of geometrical structures, background trends, noise processes and flaw types.

In developing the present scheme, an attempt was made to identify image features common to most of the radiography applications. A significant amount of research in this area has already been performed by the Image Processing Group in the Electrical Engineering and Computer Engineering department at Iowa State University. The common image features identified include flaws, noise, background trends, and variations due to geometrical structure. A large portion of their work has been focused on developing

processing routines capable of enhancing flaw features and reducing the effects of noise and background trends.

After identifying various techniques useful in reducing unwanted image features, the next step involved formulating specialized flaw detection approaches that utilized these techniques and result in the accurate interpretation of flaws in a particular class of images. Information gained from studying several different approaches that work well for different types of images was then used to develop a more generally applicable scheme. Evaluation of the different approaches resulted in grouping images based on the features they contained and identifying the processes that sufficiently reduced the unwanted features in each group. The evaluation also revealed the commonality in locating and extracting the flaw after unwanted features had been sufficiently reduced. This work provided a basis for proposing an initial scheme.

3.6 Description of Automated Flaw Detection Scheme

The scheme presented here has evolved into a two stage approach. The first stage, General Processing, performs the processes deemed necessary in reducing the contributions from unwanted features within the image. These are undesirable features common to radiographic images that tend to confuse flaw extraction and identification algorithms. The later stage, Specific Processing, is tailored for each application and performs flaw detection and classification procedures by incorporating information concerning flaw characteristics relevant to that particular application. A flow diagram of the automated flaw detection scheme is shown in Figure 3.1.

The general processing stage represents the most complex stage since it must identify and sufficiently reduce unwanted image features from the radiograph without influencing

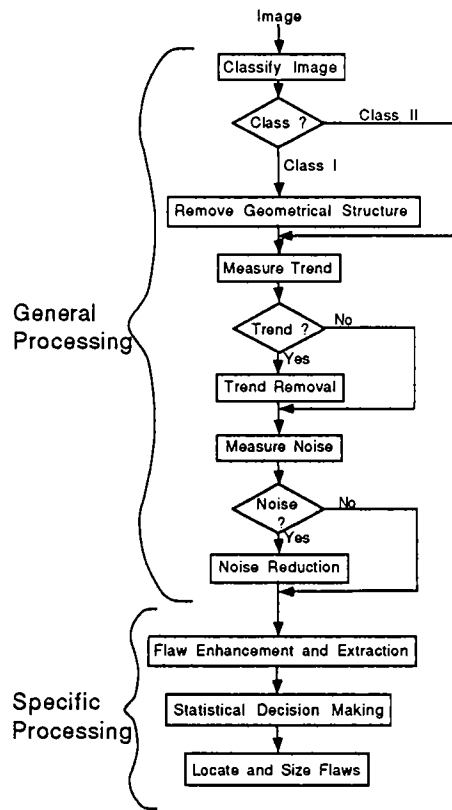


Figure 3.1: Flow diagram of proposed automated flaw detection scheme.

flaw characteristics or introducing flaw-like artifacts. For this reason, the general processing stage is the most crucial stage in the development of the general automated flaw detection system. Therefore, the focus of this thesis is on the automated feature reduction processes in the general processing stage.

The general processing stage shown in Figure 3.1 contains a subprocess for each of the three unwanted image features discussed in section 2.3; background trends, geometrical structure, and noise. Each subprocess performs measurements on the image to determine a features presence in the image. Based on these and other measures of the image characteristics, each subprocess then proceeds in reducing the effects of that

feature using the most efficient and effective technique. Image processing techniques addressing the reduction of noise and background trends in images have been researched and developed for many years. Therefore, the work presented in these areas focuses on automated identification of these features and selection of the appropriate techniques and corresponding parameters required to automatically reduce them. Although techniques for reducing the effects of noise and background trend in images are well established, the development of techniques to identify and reduce the intensity variations resulting from geometrical structure has not been researched as thoroughly. Therefore, a significant portion of the research presented in this thesis will cover the topic of developing a technique to accurately identify and reduce the effects of geometrical structure in NDE radiographic images.

The proposed automated flaw detection scheme introduced here and covered in more detail in Chapters 4-7 represents an initial scheme that will evolve into a more surely founded scheme as continued research develops improved processing techniques for industrial image interpretation applications.

CHAPTER 4. AUTOMATIC NOISE REDUCTION

4.1 Introduction

Noise is present to some degree in every radiographic image. Comments on the problem of noise in images are given in this Chapter and some successful noise reduction techniques are reviewed. Research performed in measuring the noise and determining the appropriate noise reduction technique are presented in Section 4.4. Automated noise reduction results are discussed and followed by a summary of the process and suggestions for future work in the area of automated noise reduction.

4.2 Noise Reduction Problem

Noise in a digitized radiographic image can originate from many different sources. Significant amounts of system noise can be introduced by low quality Vidicon and CCD cameras, lenses and inadequate film illumination devices [9]. Other noise sources are associated with the physical process involved in using film. This type of noise is signal dependent and can be described by a Poisson statistics model. Each finite film location is controlled by this model. In the Poisson model, the standard deviation of the signal is proportional to the square root of the average signal. Additional noise may be introduced to the film due to the specimen's material characteristics.

No matter the source of the image noise, it has the effect of reducing the detection performance of the system. Therefore, the goal of the noise reduction subprocess is

to automatically reduce the effects of noise within the image to improve detection performance. This can be a very difficult task for the system when the SNR is small and the flaw is barely visible to the human eye. In these cases, the automated noise reduction subprocess may have difficulty improving the detection performance and might degrade the performance by removing the flaw signal altogether. The automated noise reduction subprocess works reliably when the SNR is larger and the flaw signal varies slower than the noise. This is the case for the images shown in Figure 4.1. The image in Figure 4.1a is a radiographic image of a weld and the image in Figure 4.1b is a radiographic image of a railroad frog. The weld image has an overall SNR of 1.27 and the frog image has an overall SNR of 8.5. These SNR's were found by visually determining the flaw signal region and calculating the power of the signal in this region and dividing by the variance outside this region

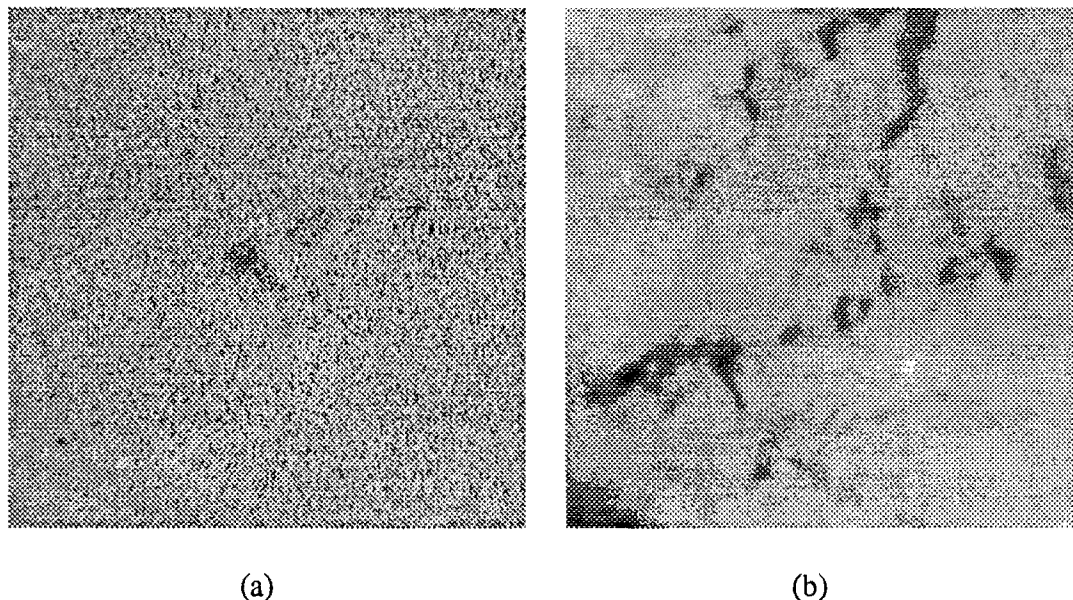


Figure 4.1: Examples of digitized radiographic images possessing different noise characteristics: (a) weld area (Courtesy of Martin Marietta); (b) railroad frog (Courtesy of American Association of Railroads)

4.3 Successful Noise Reduction Techniques

Several applicable noise filtering techniques were reviewed in Section 2.5.2. Most of these technique require few parameters, and thus are more easily incorporated into an automated noise reduction process. Filtering techniques providing a number of adjustable parameter will often yield better results when optimized for a particular image, but these filters require excessive human interaction and therefore would not be as beneficial to an automated inspection system. Conversely, nonparametric filtering techniques tailored for a less general application may not perform as well, but are more desirable due to their robust nature.

Recent research in the area of noise filter performance has been conducted by Brown [5]. Brown's work compared the performance of various noise filters in improving the detectability of the position of a step edge in an image corrupted by noise with various characteristics. The study included performance comparisons between the median, root, sigma, and adaptive smoothing filters among others. The median filter was found to be a consistently better performer in his study. Although a step edge corrupted by noise is not an accurate representation of the flaw signals and noise signals found in radiographs, the results of the study can be found useful in identifying successful noise reduction techniques.

Radiographs containing high to moderate noise levels were used in this study to test the performance of several noise filtering techniques. The filters used in the test required few input parameters. The robustness of the filters was tested by using two different images, one with a void type flaw and one with a crack type flaw, since some filtering techniques tend to work well on one flaw type and poor on the other. The ability of the noise filters to perform sufficiently well on both flaw types is crucial since

prior knowledge concerning the possible flaw types may not always be available. The noise filters were compared on the basis of their overall performance in improving the detectability of flaws in the radiographic images. The tested filters included the median, an adaptive smoothing filter [24], and a modified adaptive smoothing filter (MASF). Various window sizes were tested for each of the three filtering techniques. The MASF resulted from studying the techniques of existing filters and improving upon them.

Figure 4.2 shows the two test images used to evaluate the filters. The image on the left was produced by superimposing an actual radiographed crack on a noisy background and the image on the right, containing a void, is a subimage extracted from Figure 4.1a. The dimensions of the crack image are 100x100 pixels and the void image is 120x100. Table 4.1 lists the standard deviation of the noise and the signal-to-noise ratios for these two images.

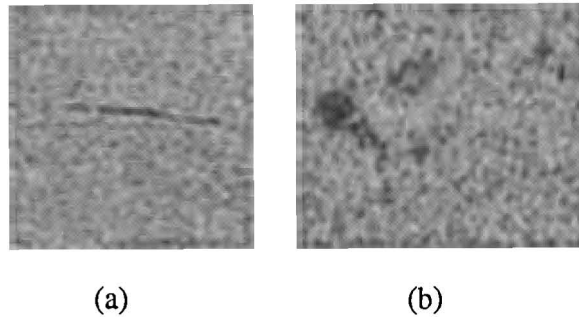


Figure 4.2: Radiographic images used to evaluate noise filter performance:
(a) crack type flaw; (b) void type flaws

The median filter's performance was tested using windows with 3, 5, 7 and 9 pixels on a side. The adaptive smoothing and MASF filters were tested with square windows of dimensions 3, 5, and 7 pixels. Figure 4.3 shows the noise filtering results for the void type flaws. Starting from the left, the top row shows the results of the adaptive smoothing

Table 4.1. Standard deviation of the noise and SNR's for test images in Figure 4.2

	Crack	Void
Standard deviation of noise	2.08	8.04
Signal-to-noise ratio	2.36	1.13
Signal-to-noise ratio (dB)	8.57	1.25

filter using window sizes of 3, 5, and 7, respectively. The last result in the top row is that of the MASF using a 3x3 window. The results using 5x5 and 7x7 windows were not significantly different and therefore are not shown here. The second row contains the residuals found by subtracting the filtered image from the original. The residual images correspond to the filtered image lying directly above them and provide a valid comparison among the other residual images within Figure 4.3. The third row contains the median filtered images using windows sizes of 3, 5, 7, and 9, respectively. The corresponding residuals are found in the bottom row.

A visual comparison of the filtered images can lead to a variety of conclusions each of which may be based on a subjective criterion. Table 4.2 provides a quantitative assessment of the noise filtered results in Figure 4.3. This table lists the standard deviation (σ) of the noise remaining in the filtered images. Based on the figure, one would subjectively conclude that the 7x7 and 9x9 median filters along with the MASF yielded superior results. This conclusion is supported by the table which shows SNR's of 3.30, 4.31 and 4.51 and σ 's of 2.48, 2.08, and 3.01 for the 7x7 and 9x9 median and MASF filters, respectively. This conclusion is further verified by the automated thresholding results shown in Figure 4.4. These thresholded images were found using Otsu's automatic

threshold selection algorithm [31] for the corresponding filtered images in Figure 4.3.

Otsu's algorithm selects a global threshold based on the images histogram. This is an example of the type of feature discrimination and extraction techniques that would be used in the Specific Processing stage to identify candidate flaws in the post-processed image. Additional processing would involve locating and sizing of the larger connected groups of pixels in these thresholded images. The flaw regions in the thresholded images corresponding to the three superior filters could be easily identified in this manner.

Figure 4.5 shows the noise filtering results for the crack type flaw. As before, the top row shows the results of the adaptive smoothing filter using window sizes of 3, 5, and 7, respectively. The last result in the top row is that of the modified adaptive smoothing

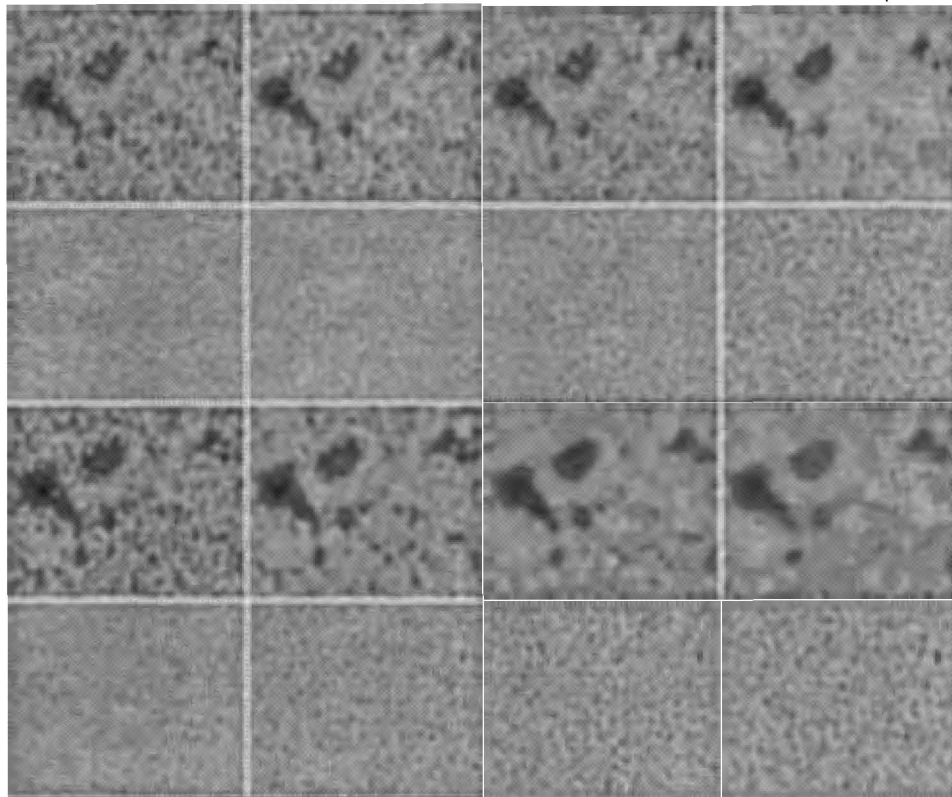


Figure 4.3: Noise filtering results for void type flaws

Table 4.2. SNR and standard deviation of noise in filtered results for void type flaws

	Original	ASF			MASF	Median filter			
		3x3	5x5	7x7		3x3	5x5	7x7	9x9
SNR	1.13	1.47	1.71	1.82	4.51	1.54	2.30	3.30	4.31
σ	8.04	5.44	4.81	4.63	3.01	4.68	3.15	2.48	2.08

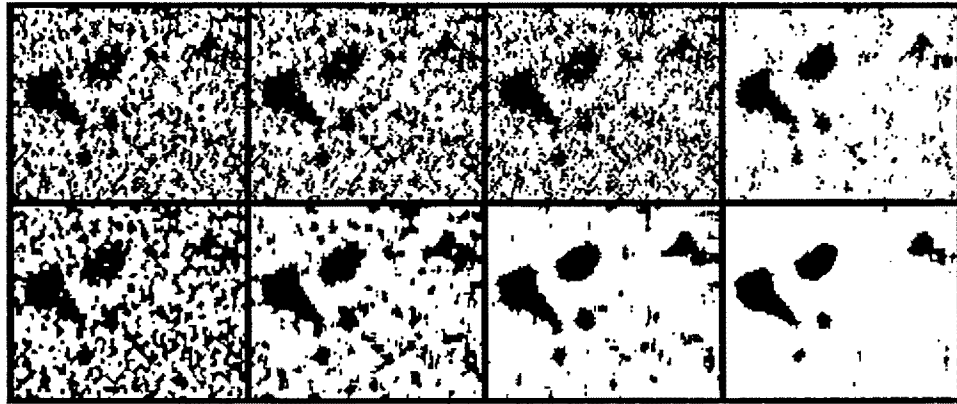


Figure 4.4: Automated thresholding results for filtered images

filter using a 3x3 window. The second row contains the residuals found by subtracting the filtered image from the original. The residual images correspond to the filtered image lying directly above them and provide a valid comparison among the other residual within the figure. The third row contains the median filtered images using windows sizes of 3, 5, 7, and 9, respectively. The corresponding residuals are found in the bottom row. The 3x3 ASF, the MASF, and the 3x3 median filter were the only filters that adequately preserved the flaw's features while sufficiently reducing the noise content. The other filters removed a significant portion of the flaw signal as can be seen in the residual images. This can be quantitatively observed from Table 4.3 where the SNR's have

become worse in all cases except those mentioned above. The outstanding performance of the MASF in this situation is noted in the table which lists a SNR of 14.66 for the filtered image. Its performance is further verified by the automated thresholding results shown in Figure 4.6. Again, these thresholded images were found using Otsu's automatic threshold selection algorithm [31] for the corresponding filtered images in Figure 4.5. The automatic flaw detection system would have a very difficult time identifying flaws in all of these thresholded images with the exception of the MASF result.

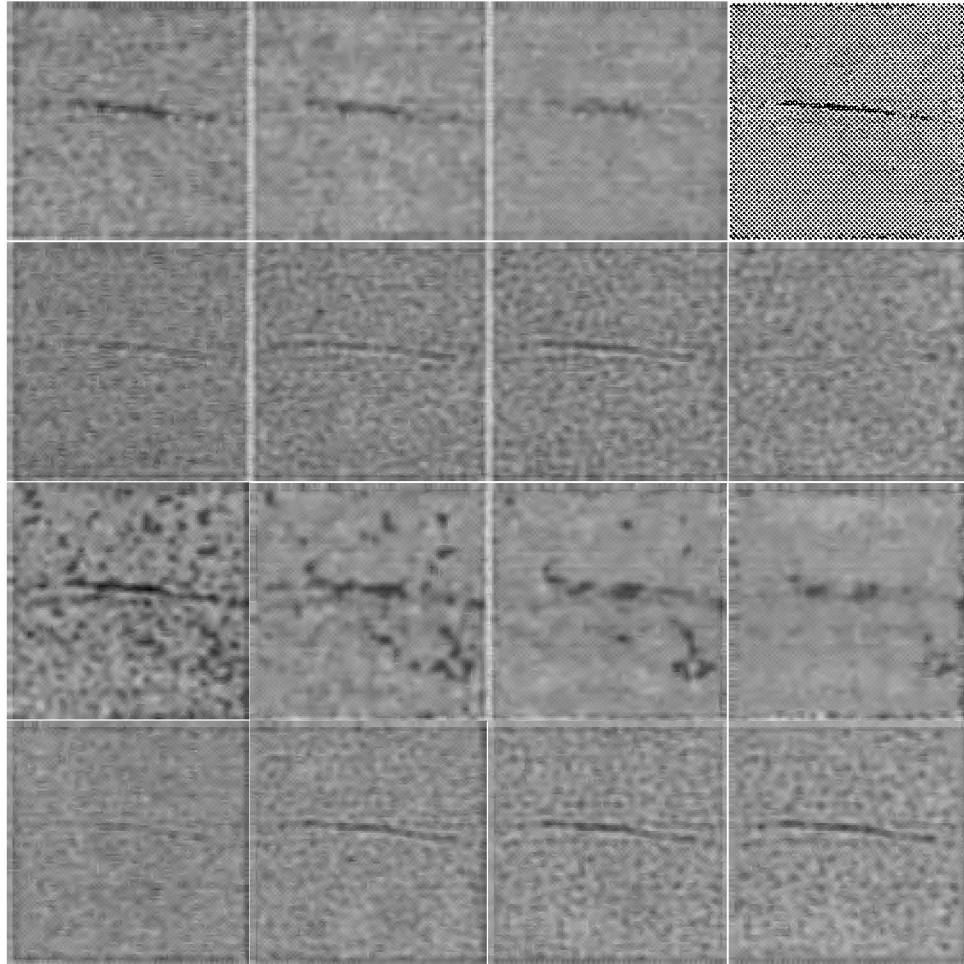


Figure 4.5: Noise filtering results for crack type flaw

Table 4.3. SNR and standard deviation of noise in filtered results for crack type flaw

	Adaptive Smoothing filter			MASF	Median filter			
	3x3	5x5	7x7	3x3	3x3	5x5	7x7	9x9
SNR	3.15	2.10	4.43	14.66	3.20	1.32	0.93	0.85
σ	1.43	0.87	0.55	0.78	1.08	0.77	0.65	0.58

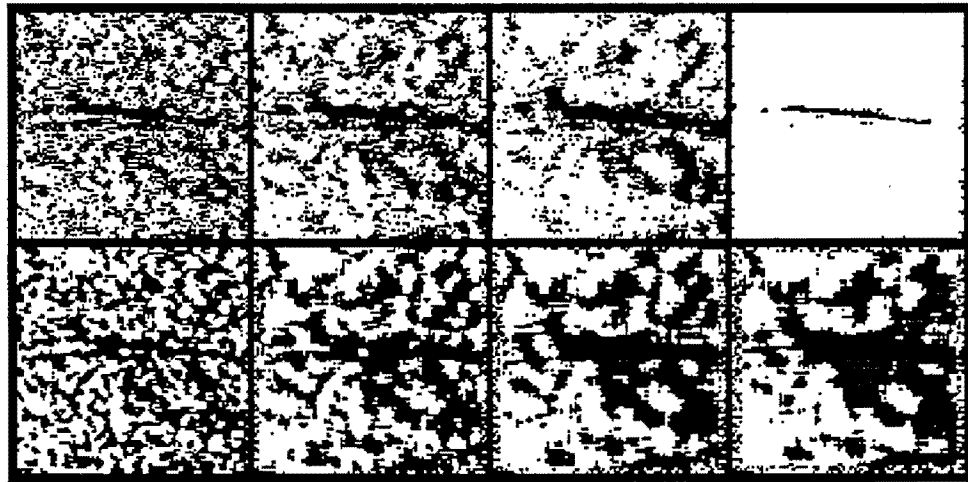


Figure 4.6: Automated thresholding results for filtered images

In comparing the performance of these filters based on the improved detectability of the two flaw types after filtering, the modified adaptive smoothing filter was found to yield the best overall performance. Although the median filter may have produced slightly better results with the test image containing voids, it failed to preserve the flaw signal in the image containing the crack. It should also be noted that the processing time involved for the adaptive smoothing filters is an order of magnitude greater than the median filters processing time. The 3x3 modified adaptive smoothing filter requires

about 40 seconds of processing time when filtering a 256x256 floating point image using a DEC 5000, whereas the 7x7 median requires a little over 1 second of processing time. This presents a trade-off between processing time and performance that may not be significant, depending on the particular application.

4.4 Automated Identification and Measurement of Noise Characteristics

As mentioned in Section 3.3, the automated noise reduction subprocess requires intelligent decision making procedures that can automatically decide whether noise reduction is necessary and if the filtering techniques improved detection performance. These decisions are based on image measurements that provide the necessary image information.

Appropriate image measurements were developed by first visually identifying the noise characteristics within the images and then setting out to quantitatively describe them via numerical image measurement models. To develop a measurement for determining when noise reduction is necessary, several images containing undesirable noise characteristics were selected. Then, based on these images, a technique was developed to objectively measure the noise level. Many times the proposed measurement was not consistent in fully describing the noise content or was not able to properly discriminate between the noise and other image features. The noise measure developed here consistently yields an objective quantitative value for the noise level of the images studied here.

The proposed measure is based on the local variance calculations within an image. The measurement is found by first calculating the local variance for each non-overlapping $n \times n$ window in the image (8x8 windows used here). The square root of these sample variances (σ) are then normalized by dividing by the intensity range of the image data.

Next, the average and standard deviation of the normalized σ 's corresponding to the non-overlapping windows is calculated. The noise level is then calculated by averaging together the normalized σ that are less than the average σ plus two standard deviations. This is done to help reduce the higher local variance bias due to fluctuating flaw signals. The resulting value is then multiplied by ten for ease of handling.

Other noise measures were found useful in characterizing the noise after processing. One measure, referred to as sigma of the noise, is the standard deviation of the unsharp masking result. Recall that unsharp masking is performed by subtracting a 3x3 boxcar smoothed image from the input image. The other measure is referred to as the zero crossing count. This measure involves counting the number of zero-crossings that occur in the first derivative of the selected rows and columns. The number of zero-crossings are calculated from 10 uniformly spaced rows and columns in the input image and a 3x3 boxcar smoothed version. The zero-crossing measure is then found by subtracting the total count found for the smoothed image from the original image. These measures are useful in evaluating the success of the noise filtering technique applied.

The average local variance (ALV) measure was calculated for the images shown in Figure 4.7 which is the source of the void type flaw test image mentioned earlier. In this figure, the upper left image is the original, the upper right is the adaptive smoothing filter result, the lower left was filtered with a 3x3 MASF and followed by a 3x3 median to help remove impulse-like noise, and the lower right image is the result of a 7x7 median filter. The noise measures calculated for these images are found in Table 4.4. Applying a 3x3 median filter improved the ALV measure very little in these particular images. However, the zero-crossing measure can be very useful in identifying situations when a 3x3 median filter would be helpful in reducing impulse type noise. This is evident in

both of the following tables.

Table 4.4 illustrates the consistency of the noise measure to objectively quantify the image noise level within the group of images shown in Figure 4.7. Through visual inspection, one notices the correlation between observed noise content and calculated noise level. To gain a better feel for the noise and flaw signal fluctuation found in these images, a row of data was extracted from each of the images listed in Table 4.4 and

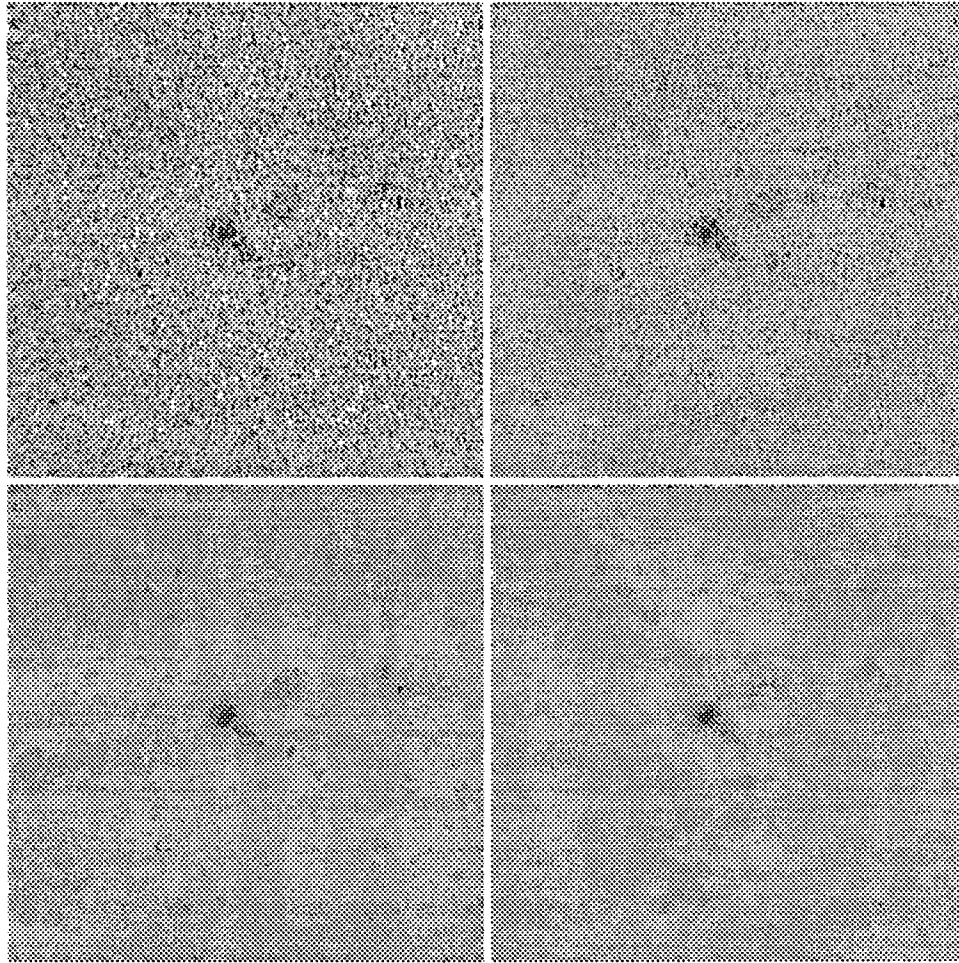


Figure 4.7: Images used to illustrating ALV noise measure; clockwise from the upper left corner is the original, the adaptive smoothing filter result, 7x7 median filter result, and the result of the 3x3 MASF followed by a 3x3 median

Table 4.4. Noise measures found for images in Figure 4.7

	Original	Adaptive Smoothing Filtered (ASF)	(3x3) Median of ASF	Modified Adaptive Smoothing (MASF)	(3x3) Median of MASF	(7x7) Median Filtered
ALV	7.236	6.313	5.778	4.416	4.380	4.226
Std. Dev. of noise σ	6.783	6.304	3.569	4.391	2.947	4.110
Zero crossing	2.637	2.875	1.484	3.557	1.337	1.680

plotted in Figure 4.8. The data plotted here intersect the darkest portion of the largest flaw. The same row position was extracted from each of the images. Starting at the top of the figure, the first slice is from the original image, the remaining slices are from the ASF, MASF, 3x3 median of MASF, and 7x7 median filtering results, respectively.

The plotted slices provide a good comparison for the effectiveness of the filters in reducing the noise and preserving the flaw signal. The inherent noise suppression abilities of the filters are more easily perceived from the plots than the signal preservation characteristics. When closely examined, one notices differences in the magnitude of the filtered flaw signals which lie between 100 and 130 on the horizontal axis. The adaptive smoothing filter and the median filter reduced the magnitude of the flaw signal significantly, compared to the original. However, the MASF performed exceptionally well in both preserving the flaw's characteristics and suppressing the noise.

Another example of images quantified by the ALV measure is shown in Figure 4.9. In this figure, the upper left image is the original, the upper right is the adaptive smoothing filter result, the lower left was filtered with the modified adaptive smoothing filter and followed by a 3x3 median to help remove impulse-like noise, and the lower right image

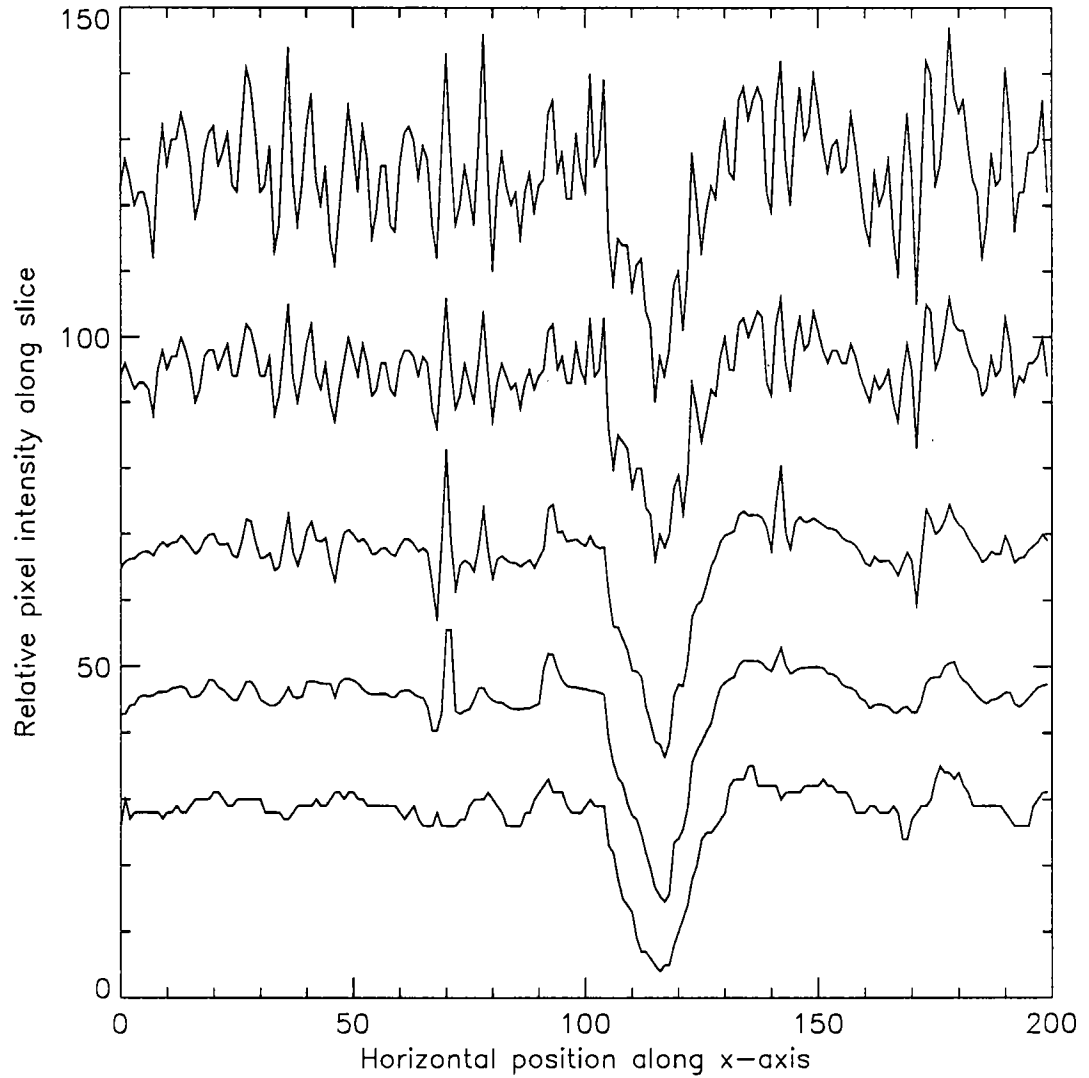


Figure 4.8: Slices of image data illustrating noise and signal suppression for different noise filtering techniques; plot shows same row position; the top slice is from the original image, the remaining slices are from the ASF, MASF, 3x3 median of MASF, and 7x7 median filtering results, respectively

is the result of a 7x7 median filter. The noise measures calculated for these images are found in Table 4.5.

The ALV noise measure has been found to be a practical noise measurement tool in application to the automated noise reduction identification process. The ALV noise measurement was successful in accurately quantifying the image noise content when the image intensity was relatively uniform across the image. The other noise measures

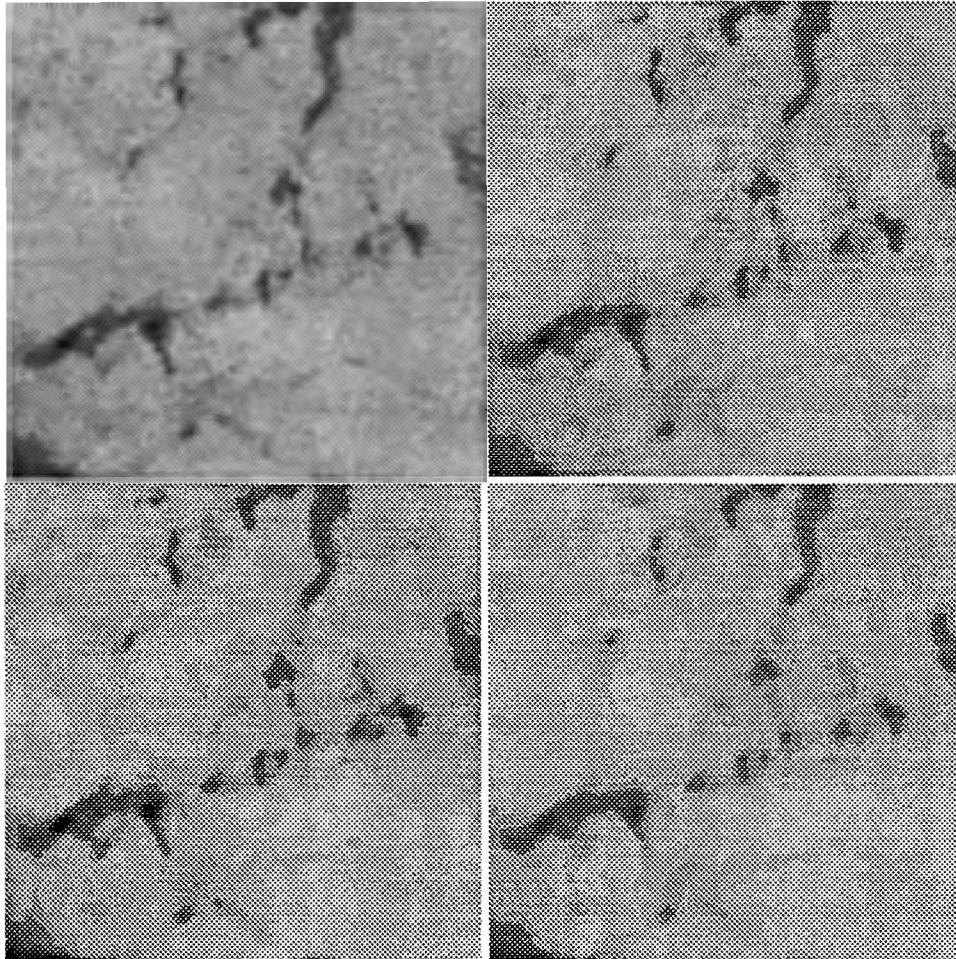


Figure 4.9: Images used to illustrating ALV noise measure; clockwise from the upper left corner is the original, the adaptive smoothing filter result, 7x7 median filter result, and the 3x3 MASF result

Table 4.5. Noise measures found for images in Figure 4.9

	Original	ASF	(3x3) Median of ASF	MASF	(3x3) Median of MASF	(7x7) Median Filtered
ALV	5.570	3.504	3.189	3.328	3.030	3.599
Std. Dev. of noise σ	3.115	1.514	0.668	1.259	0.634	0.928
Zero crossing	3.559	3.871	1.555	3.996	1.305	1.336

studied here were not found as effective in accurately quantifying the noise content. The σ noise measure was not sensitive to the “grainy” type of noise observed in the weld image in Figure 4.7. The zero crossing count was also ineffective in detecting this type of noise, but was found to be quite sensitive to small impulse type noise that can be sufficiently reduced by a 3x3 median filter.

After developing reliable noise measurement tools, decision rules for the automated noise reduction subprocess must be established. The first decision in the noise reduction subprocess is to determining if noise reduction will improve the flaw signal detection performance. This decision is based on the ALV noise measure results. Based on experimentation, a proper threshold level can be selected to determine when noise filtering would most likely improve detection performance. The second decision in the noise reduction process is to evaluate the noise filtering result to determine if it was sufficient in reducing the noise. This can be done by comparing the ALV and other noise measures before and after filtering. The noise reduction subprocess can be easily modified to accommodate additional noise measures and filtering techniques relevant to the application at hand.

4.5 Automated Noise Reduction Results

In addition to the noise filtering results shown in previous sections where the proposed noise reduction techniques were evaluated, Figure 4.10 shows an example of a simulated image containing a large range of noise levels superimposed on a nonstationary background. As mentioned before, the ALV noise measure is reliable only when the input image has uniform intensity across the entire image. However, the modified adaptive smoothing filter introduced earlier and found to be superior to other noise filter techniques is a reliable noise reduction technique in images possessing nonuniform intensities. This is illustrated in Figure 4.10. The first row shows the input image corrupted by decreasing noise levels. The second row shows filtering results from the modified adaptive smoothing filter. The bottom row show filtering results from a 7x7 median filter. The SNR's corresponding to these images are recorded in Table 4.6. As observed in the images and recorded in the table, the MASF and Median filters did not improve the calculated SNR for the 1st image. However, they did perform rather well in suppressing the noise content and preserving the flaw signal in the 3rd and 4th images. The MASF was the superior performer in preserving the sharp discontinuities. The program listing for the modified adaptive smoothing filter can be found in Appendix A.

The automated noise identification and reduction process described here is reliable in quantifying the noise content in images containing relatively uniform intensities and has demonstrated its robustness in sufficiently reducing the noise characteristics present in the range of radiographic images presented here. Future work in the area of automated noise reduction would benefit from the development of additional noise measurement tools capable of reliable noise quantification in images possessing nonuniform intensity variations across the image. This work could be based on the radiographic image

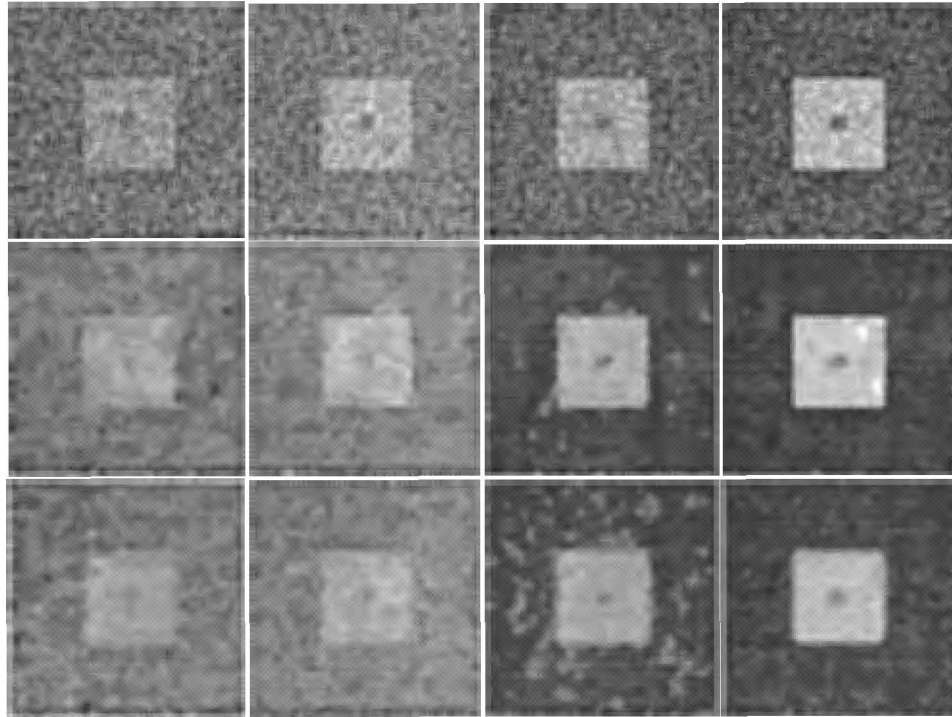


Figure 4.10: Nonuniform images containing a large range of noise levels which are listed in Table 4.6

Table 4.6. SNR's for images shown in Figure 4.10

	1 st	2 nd	3 rd	4 th
Originals	0.963	1.17	1.41	1.56
3x3 MASF	0.660	3.42	6.05	11.15
7x7 Median	0.693	2.13	3.96	6.52

formation model and result in more accurate identification of the image's Poisson noise characteristics.

CHAPTER 5. AUTOMATIC TREND REDUCTION

5.1 Introduction

Background trends are present in many NDE X-ray images. This chapter focuses on the problem of trends in images and reviews some successful trend-reduction techniques. Research performed on identifying trend characteristics and on developing an automated trend removal subprocess is presented in Section 5.4. Some trend reduction results are then presented and followed by suggestions for future work in the area of automated trend reduction.

5.2 Trend Reduction Problem

As mentioned in Section 2.3, trends are global changes in image intensity due to nonuniform variations in material density or thickness. These background trends are represented in digitized radiographic images as smooth continuous variations in image intensity. In this form, they can be addressed by digital image processing algorithms. These slow continuous variations introduce nonstationarity and make it difficult to accurately segment the flaws from the background. Images containing a large intensity range, attributed to large variations in the specimens material thickness or density, tend to hide the minute details associated with a small flaw signal embedded in the trend. Furthermore, reliable histogram-based segmentation techniques used to identify candidate flaws are thoroughly confused by background trends and yield unacceptable results.

Radiographic images containing trends are common in industrial radiography. Therefore, trend reduction is a necessary step in preparing a radiographic image for automatic flaw detection algorithms.

Examples of images with these characteristics are found in Figure 5.1. The top image in this figure shows a radiograph of an oil pipe containing a crack. The trend in this image is mostly vertical in that the film density along the top of the radiograph is low and steadily increases to a higher film density along the bottom. The middle radiographic image is from a composite material. The trend varies slowly down the image until close to the bottom of the image where a sharper transition in material density or thickness yields an abrupt increase in the film density. The bottom radiographic image is from a railroad frog. The trend in this image varies in intensity in a diagonal fashion from the upper left-hand corner to the lower right. This trend is rather subtle but still presents problems to the automated flaw detection programs.

As seen in Figure 5.1, background trends can take on different shapes and contain swift or gradual changes in intensity across the image. Most often the background trend characteristics can be modelled by 1st and 2nd order polynomials or step-like transitions. These assumptions are adequate in describing the global film density variations most often encountered in the industrial radiographic images studied here.

5.3 Review Trend Reduction Techniques

Several techniques have been developed to effectively reduce background trends without modifying the flaw signal or introducing additional artifacts. These techniques include bivariate polynomial surface fitting discussed in Doering [11], one-dimensional polynomial fitting [11], and mathematical morphology discussed in Chackalackal and

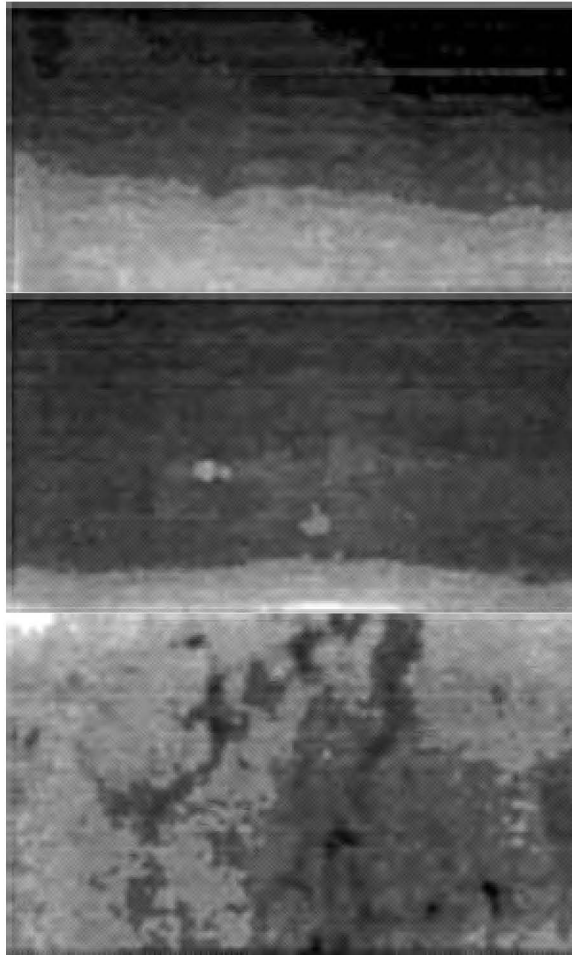


Figure 5.1: Examples of radiographic images containing background trends: Courtesy of Atlantic Richfield, Westinghouse, and American Association of Railroads

Basart [6] and Skolnick [41]. All of these techniques have been found to yield similarly useful results. However, the techniques do differ in the amount of processing time involved and the type of a priori knowledge necessary to produce successful results.

Surface fitting techniques are computationally intensive and require substantial processing time. Both one-dimensional polynomial fitting and morphology are relatively quick methods. Chackalackal and Basart [6] have shown mathematical morphology to

be a powerful tool in this area, but the technique requires prior knowledge of the flaw's shape, size and orientation in determining the appropriate structuring element.

The technique found most suitable in reducing the background trends was one-dimensional polynomial fitting. In addition to the polynomial model, one other assumption applies. The maximum flaw width must be less than one-fifth of the images dimensions to avoid remove the flaw signal using polynomial fitting. This is a reasonable assumption for the majority of the NDE X-ray imaging applications. The polynomial fitting technique is very fast and effective in reducing the trends encountered most often in NDE X-ray images. Furthermore, this technique requires no a priori knowledge concerning the flaws characteristics.

Polynomial fitting is performed on the individual rows or column within the image after determining the appropriate polynomial order and deciding whether to fit in the row or column direction. Once the parameters are selected, a least squares approximation of the background trend is found for each row or column and subtracted from the corresponding row or column to yield a residual image. This residual image represents the variations due to flaw signals and noise and can be successfully processed by noise filters and traditional histogram-based segmentation techniques to identify candidate flaws.

5.4 Automatic Identification and Measurement of Trend Characteristics

Automation of the one-dimensional polynomial fitting technique involves the identification of the trend characteristics and the parameters required for successful reduction. The trends characteristics can be identified by developing measurements based on the allowable background trend characteristics mentioned above.

Two measures were developed to identify the trend and determine the appropriate fitting parameters. The first measure is used to determine if adding an offset to each individual row and column would be sufficient in reducing the background trend. This is achieved by calculating two values representing the variation in the background for both the rows and columns. Each value is found by selecting ten uniformly spaced rows and columns within the image and calculating the slope of the least squares line for each of the ten rows and columns. The variance of the ten slopes for each direction is then calculated. This measure is normalized by multiplying by the number of pixels in the corresponding direction. The result is a measure of the complexity of the background trend. This measurement can be more clearly described using an example.

Figure 5.2a shows a radiographic image, which will be referred to as Mart, that contains a noticeable background trend. The surface plot of Mart in Figure 5.3 provides a useful illustration of the overall trend present. Ten uniformly spaced rows and columns were extracted from Mart. The slopes of the least squares lines approximating the data in these selected rows and columns were then used to calculate the measures. Figure 5.4 shows a reconstructed plot of the approximated lines for the selected rows (a) and columns (b).

The normalized variance was 4.2 for the row slopes and 8.2 for the column slopes. A measure of normalized variance below 50 is considered low. A low variance among both the row and column slopes implies a simple trend that can be reduced using Eq. (5.1). This equation can be interpreted as a zero order fit to the image rows and columns where

$$new\ x_i = (x_i - row\ mean) + global\ mean. \quad (5.1)$$

In Eq. (5.1), x_i is the intensity value of a pixel in the current row, *row mean* is the average of all the pixel values in that row, and *global mean* is the average of all the pixel values

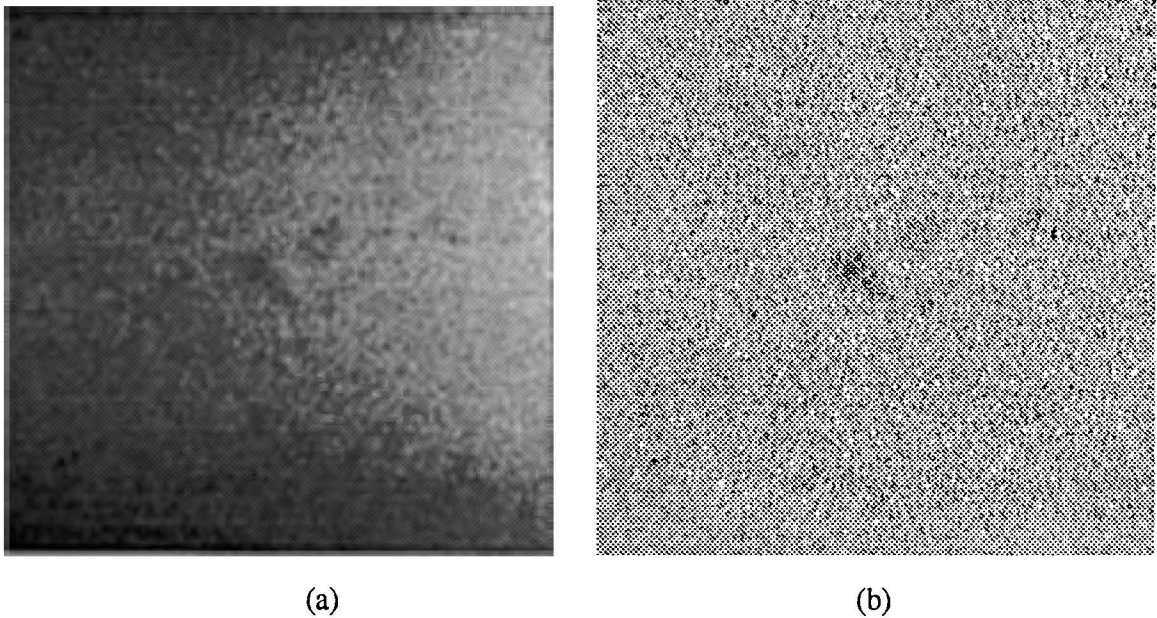


Figure 5.2: Radiographic image of a fuel tank weld (Courtesy of Martin Marietta): (a) before; (b) after trend removal

in the original image. The new intensity value for x_i , *new* x_i , is found by subtracting the *row mean* and adding the *global mean*. The zero order fit is extremely fast since it requires no matrix computation, but rather the average of the row's intensity values and a few additions. Because of its speed, this technique has been implemented by Doering in a real-time radiography system to enhance the observed image in real-time [10].

After determining that zero order fitting is applicable, two additional measurements can be calculated to determine if it should be applied to the rows, the columns or both. These measurements are found by calculating the standard deviation of the mean values calculated for the rows, and then for the columns. A high standard deviation in the selected row mean's implies the need for zero fit to the rows and the same goes for the columns. The standard deviation measured for the row and column mean's was 56.6

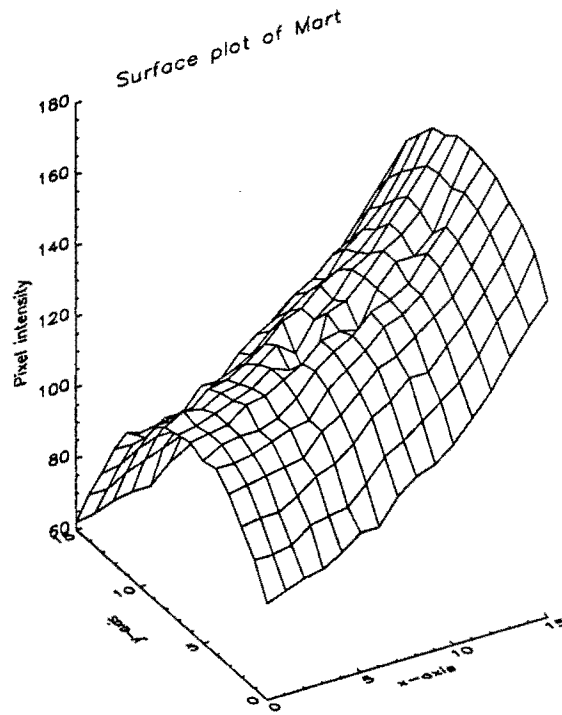


Figure 5.3: Surface plot of intensities in Mart

and 70.1, respectively. Therefore, zero order fitting was applied to both the rows and columns in Mart.

Figures 5.5 and 5.6 show intermediate results of the zero order fit for the columns and rows, respectively. As can be seen in these figures, the zero order fit is a line by line shifting process that offsets the recorded intensities in each line by a unique constant determined by Eq. (5.1).

The surface plot in Figure 5.6b is plotted with the same axis scaling that was used in the previous surface plots. This allows one to accurately compare the surface plots showing the nonuniform background trend before and after trend reduction. The surface plot in Figure 5.6b depicts the trend-reduced image shown in Figure 5.2b. The zero

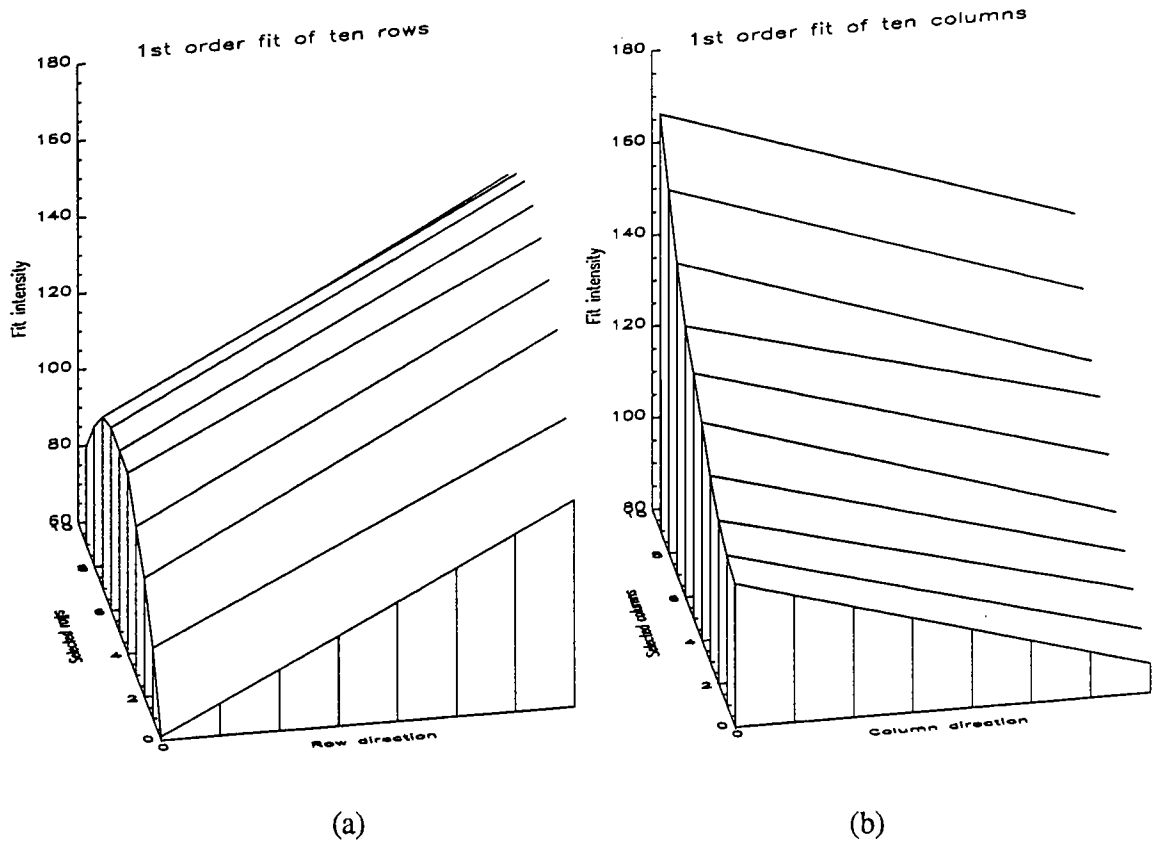


Figure 5.4: Reconstructed plot of the least squares lines calculated for the selected rows (a) and columns (b)

order trend removal technique used here has been an effective enhancement technique in many radiographic images.

When a high variance is measured among either the row slopes or the column slopes, it implies that a complex trend exists. These trends require higher order polynomial fitting. When higher order fitting is found necessary, a second measure is used to determine the polynomial order required and whether the fit should be applied to the rows or columns. This is done by evaluating the errors incurred by fitting 1st and 2nd order polynomials to

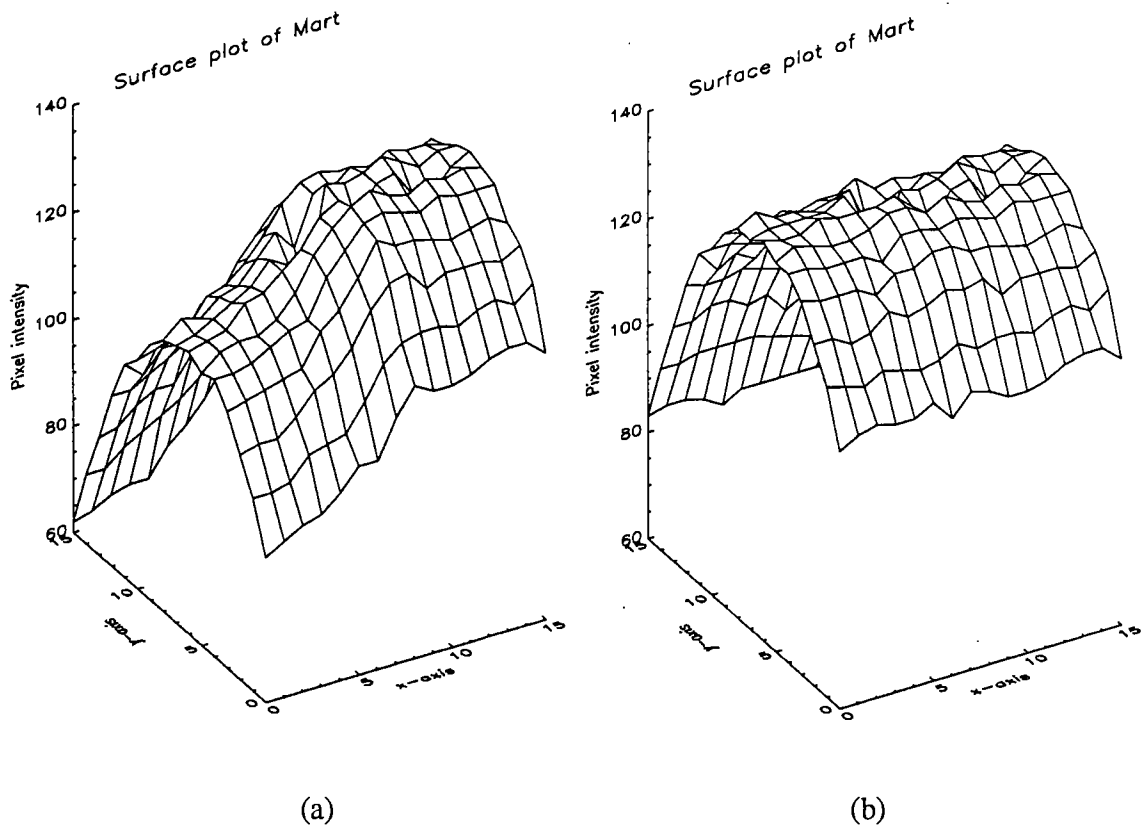


Figure 5.5: Intermediate surface plots of Mart demonstrating the zero order fitting technique: (a) half done with column fitting; (b) completed column fitting

the ten selected rows and columns. The order and coordinate direction combination that incurs the minimum error is selected as the parameter set for the trend removal process. Since 1st order fitting is faster, it is used when the error is not significantly greater than that found using 2nd order polynomials. Higher order polynomial fitting works basically the same way as the zero order fitting example with the exception that each row or column is approximated by a 1st or 2nd order polynomial which is then subtracted from the original image intensities to yield a residual image.

Based on these measurements, the automatic trend reduction diagram shown in Figure

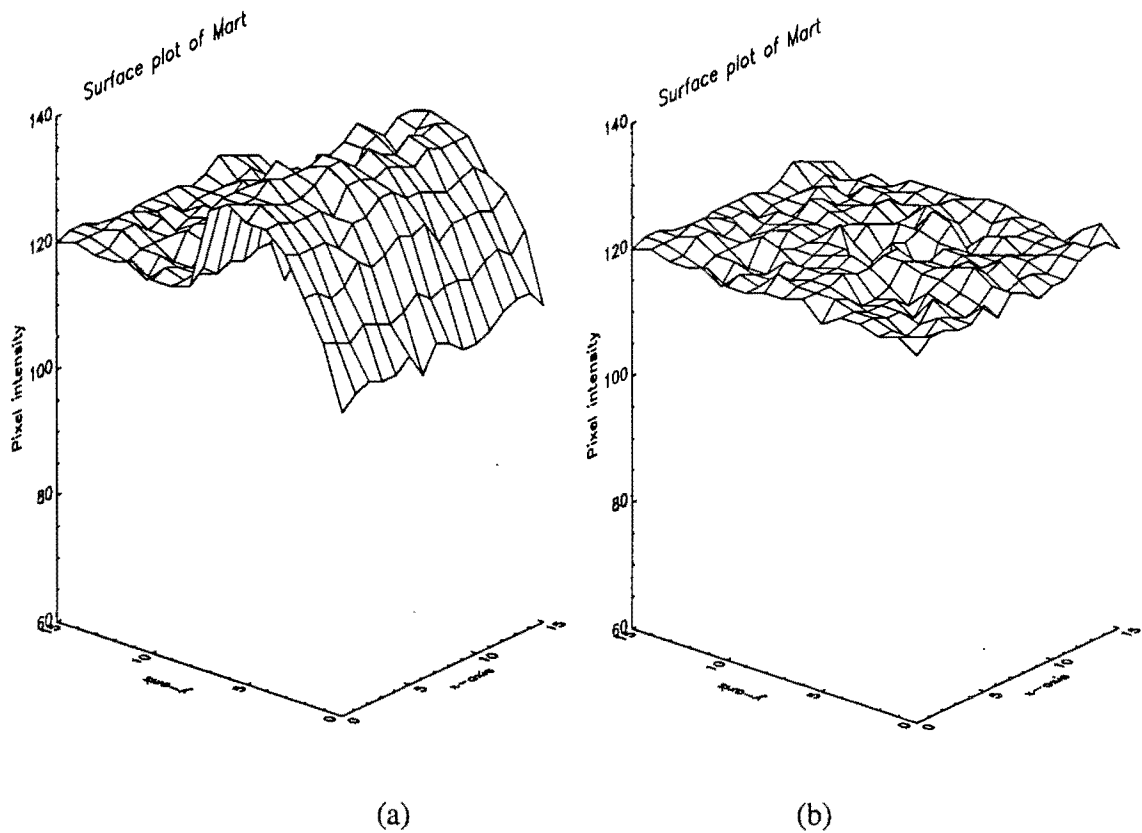


Figure 5.6: Intermediate surface plots of Mart demonstrating the zero order fitting technique: (a) half done with row fitting; (b) completed row fitting

5.7 was developed. The first block encountered by the image, *Trend Measure 1*, calculates the magnitude of the normalized slope variance measures to determine if the background trend is complex. If the variances are too large, the trend is identified as complex and the appropriate parameters are found and used to reduce the trend based on the polynomial model. If the trend is not identified as complex, the variance of the row and column means is calculated to determine if the zero order fitting technique would be useful. This is performed in the block labeled *Trend Measure 2*.

Within the trend reduction subprocess, preselected thresholds are compared with

each measurement to determine the appropriate trend reduction technique. The proposed diagram provides for situations when 1st or 2nd order polynomial fitting are found to be inadequate. In these cases, fitted polynomials poorly approximate the background trend and more powerful methods such as mathematical morphology [6, 41] are required.

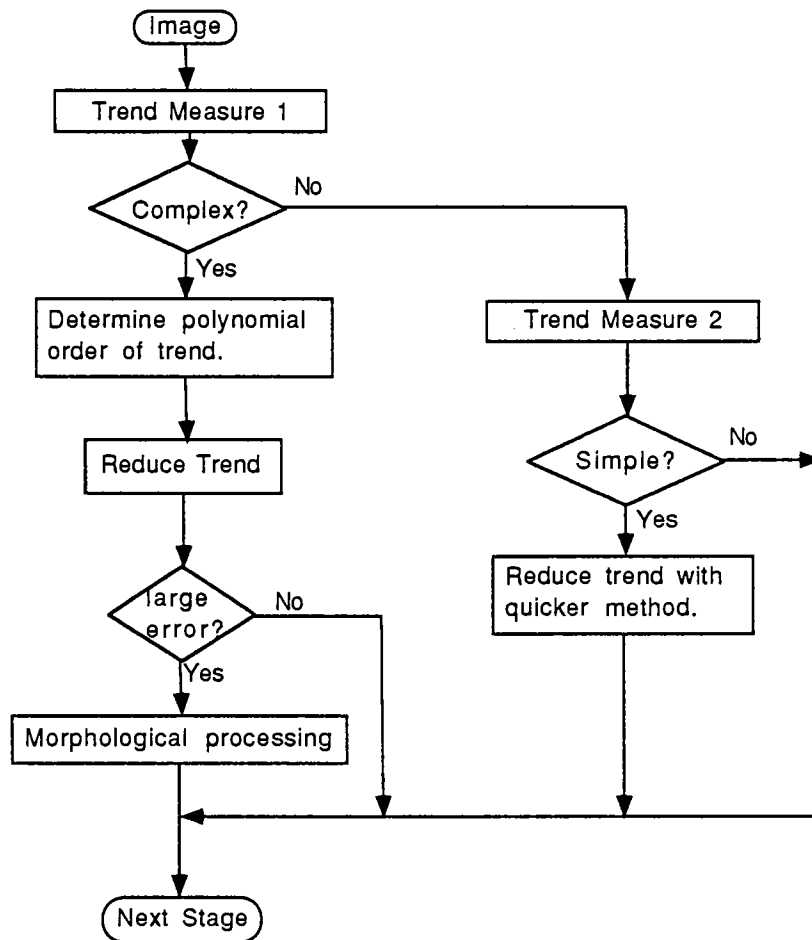


Figure 5.7: Flow diagram of automatic trend reduction subprocess

5.5 Automatic Trend Reduction Results

The two images in Figure 5.8a were processed by the automatic trend reduction procedure. When analyzing the top image in this figure, the normalized slope variance measures found for both the rows and columns were much less than the threshold of 50, therefore the trend was not identified as complex. To determine if zero order fitting was necessary, the standard deviation of the row and column means were calculated. They were found to be 63.5 and 6.7, respectively. This tells us that the background trend varies with the rows, or in the vertical direction. Therefore, the trend was removed by applying the zero order fit to the rows only. The resulting image is shown at the top of Figure 5.8b.

The background trend in the bottom image was found to be more complex. In addition to the simple method used above, 2nd order polynomial fitting to the columns was also necessary. Polynomial orders higher than 2nd order are not used since they tend to introduce artifacts [11].

Figure 5.9 shows another example of trend reduction using the automated trend removal procedure. The image in the upper left corner is the original radiographic image of a pipe weld. The image to the right is the histogram equalized version of the original. The lower left image shows the background trend identified by the automated trend reduction procedure. The result of trend removal is shown in the lower right image. Measurements from the original image suggested that 2nd order polynomial fitting should be performed on the columns of the image. The row and column slope variances were 54.5 and 53.0, respectively.

The background features in all of these images were sufficiently reduced using the automated trend removal procedure. This is visually verified by comparing the original

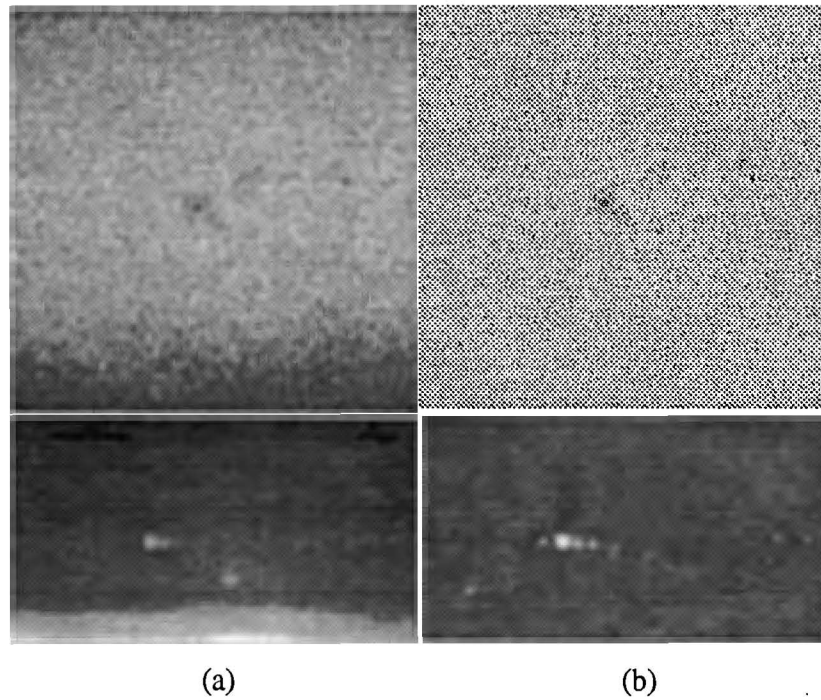


Figure 5.8: Radiographic images before and after trend removal: (a) originals; (b) results after trend removal; top image is from the fuel tank weld referred to previously, and bottom image is a composite material (Courtesy of Westinghouse)

image to the trend-reduced result and quantitatively verified by comparing the trend measures before and after trend reduction.

Although the automatic trend reduction procedure works well on many images, the introduction of prominent artifacts can occur in some situations. This is most often due to the fact that the 1st or 2nd order polynomial model does not apply to the observed background trend. In these situations, a different background trend model is required. For this reason, further work in the area of automatic trend reduction could focus on the development of a more general model or a model that more readily adapts to the observed background trend.

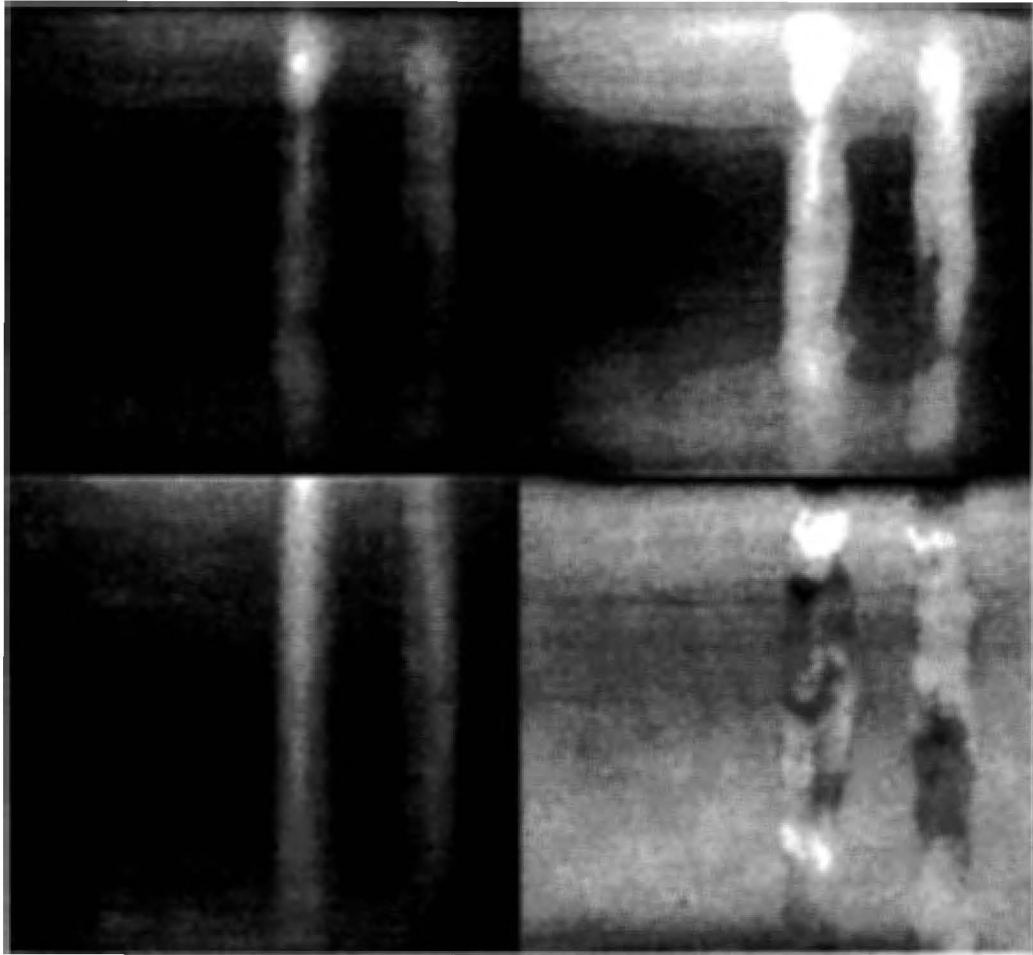


Figure 5.9: Radiographic image of pipe weld before and after trend removal:
top left image is original; image to the right is histogram equalized
version of original; bottom left is approximated background
trend; bottom right is result after trend removal

CHAPTER 6. GEOMETRICAL STRUCTURE REDUCTION

6.1 Introduction

Many NDE X-ray imaging applications involve the inspection of a specimen possessing local variations in material thickness due to the specimens' shape or geometrical structure. This Chapter discusses the problems associated with inspecting digitized radiographs that possess intensity variations due to geometrical structure and presents the research conducted in developing a technique to automatically reduce these unwanted variations. The approach to solving this problem is presented in Section 6.3. Section 6.4 presents a description of the technique along with an example for each of the main steps involved. This is followed by a more detailed discussion of the technique in Section 6.5. Some results are then presented and followed by a discussion the techniques deficiencies and suggestions for future work.

6.2 Geometrical Structure Problem

Intensity variations due to geometrical structure come in many different shapes and forms. The number of different geometrical structures is as numerous as the number of different X-ray imaging applications. Unlike background trends with global characteristics, these variations are locally defined and attributed to local changes in the material thickness or material density which are often associated with the shape of the radiographed specimen. The effects of geometric structure in the film can be minimized

by choosing the position and orientation of the object such that the resulting projection yields the least complicated geometrical structure.

Examples of variations due to geometrical structure in radiographic images are found in Figure 6.1. The structure of interest in the top image results from a section of a circular ring of thicker material. The magnitude of the local variations occurring on the outside radius boundary are approximately 100 grey levels per 10 pixels. The “T-shaped” structure in the lower left image is lying on a nonuniform background. This yields an image with a multitude of local variations possessing a large range of sharp intensity changes. The largest change is 220 grey levels over 7 pixels, however, most local variations involved 40-80 grey levels over 7 pixels. The lower right image contains a ridge of thicker material bridged across a relatively uniform background. This particular structure contains local variations in intensity of about 40-140 grey levels over 10 pixels. These are examples of typical radiographic images containing variations due to the specimen’s geometrical structure.

Like background trends, images containing a large intensity range, attributed to geometrical structure, tend to hide the minute details associated with a small flaw signal. Edge detection techniques are not able to discriminate between sharp variations in flaw signal and variations due to geometrical structure. Furthermore, normally reliable histogram-based segmentation techniques used to identify candidate flaws are thoroughly confused by background trends and yield unacceptable results. Radiographic images containing geometrical structures are common in industrial radiography. Therefore, geometrical structure reduction is another necessary step in preparing a radiographic image for automatic flaw detection algorithms.

As mentioned before, little research has been performed in NDE concerning the

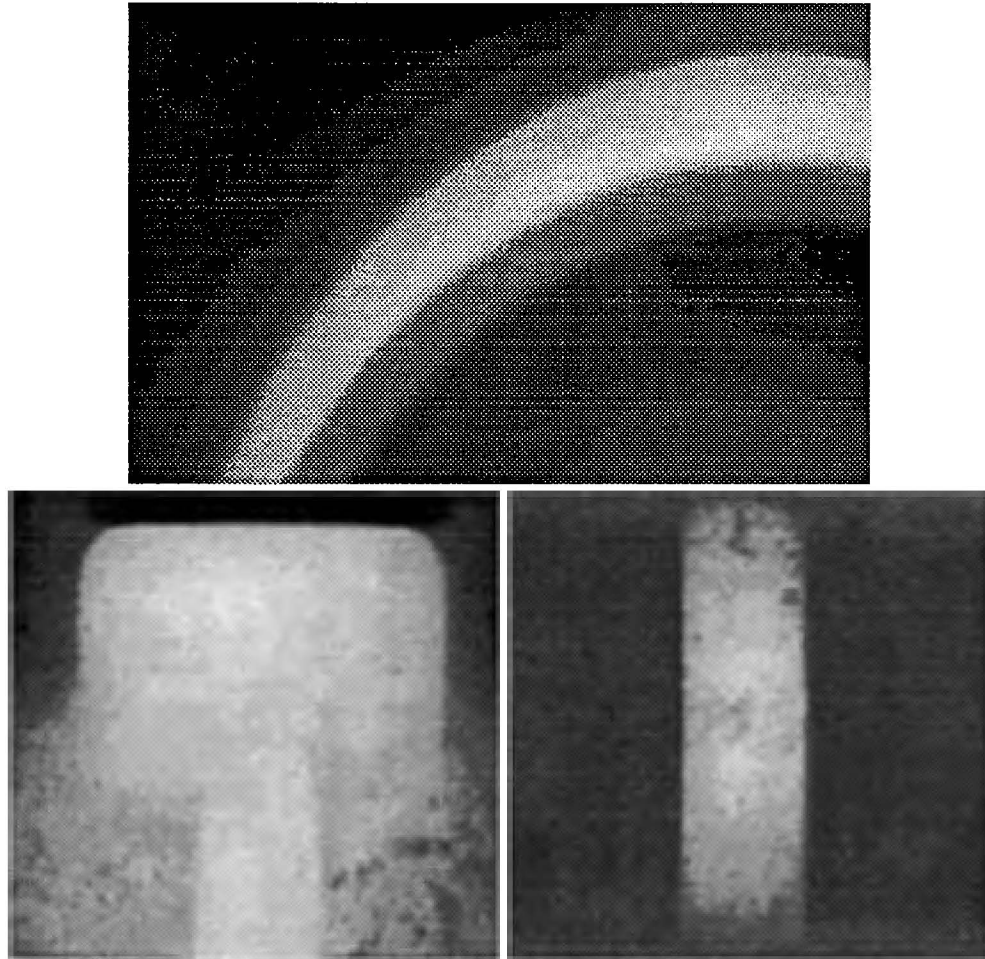


Figure 6.1: Examples of radiographic images containing geometrical background structures

estimation and removal of features due to geometrical structure. However, the image processing and pattern recognition community has researched the general problem of segmentation for many years [8, 12, 2, 42], as discussed in Section 2.5.3. Morphological processing techniques [6, 41] have been successful, but require prior knowledge of the flaw's shape, size and orientation. Part registration and subtraction techniques [40] yield promising results also, but require prior knowledge of part geometry and precise registration constraints. The author has studied the application of smoothing and

subtraction techniques to the problem and shows some results below.

Figure 6.2 shows results from some of these techniques. The original image is shown in the top left corner. The image to the right, generated by Siwek [40], is the residual found by registration and subtraction of a simulated radiographic image from an actual digitized radiographic image. The simulated image was generated by an X-ray simulation program called XRSIM [20] that requires a CAD model of the specimen. This is an ongoing area of research at Iowa State University's Center for Nondestructive Evaluation. The remaining images were generated by the author using smoothing techniques. The corresponding residuals were generated by subtracting the smoothed image from the original. The residual images are scaled and provide a valid comparison among the other residuals in the figure. The last two images in the top row are the smoothed image generated by one pass of a 6x6 Gaussian convolution mask and the corresponding residual. The first image pair in the second row is the smoothed image generated by two passes of a 6x6 Gaussian convolution mask and the corresponding residual. The next image pair includes the results from one pass of a 7x7 boxcar smoothing filter and the corresponding residual. The first image pair on the bottom row was generated by two passes of a 7x7 boxcar smoothing filter. The last image pair was generated by the modified adaptive smoothing filter.

The various smoothing techniques were studied to determine if they were capable of discriminating between variations due to flaws and variations due to geometrical structures. The techniques used above differ by the distribution of weights placed on the pixels when smoothing. The Gaussian mask places greater significance on the pixels closer to the center. This results in a smaller point spread function and thus a more accurate approximation at sharp transitions. The 7x7 boxcar places equal weights on all

pixels in the mask. This results in a larger point spread function and a greater blurring effect. Two passes of a particular filter result in a smoother image that is insensitive to locally concentrated intensity variations, such as those characteristic of flaws or sharp transitions due to geometrical structure. These items can be observed from the figure when comparing the residuals found between the first and second pass of a particular filter and when comparing the residuals produced by different masks.

The residual images reveal a prominent artifact caused by the techniques' inability to accurately approximate intensities at abrupt transitions. This is not a significant problem when inspecting the images visually. However, this is a significant problem when implementing this method in an automatic flaw detection system since the algorithms will easily mistake these artifacts for candidate flaws.

6.3 Geometrical Structure Reduction Approach

The figure mentioned above demonstrates the need for a robust image processing technique that provides a reliable estimate of the abrupt intensity variations due to an arbitrary geometrical structure and remains insensitive to flaw signals. This problem reflects the ever-present uncertainty associated with segmentation and boundary identification problems. That is, one needs to accurately identify the boundaries of the geometrical structure automatically without mistaking flaw signals for geometrical structure. The approach taken here then is to develop a technique that approximates the intensity variations due to geometrical structure by piecewise-smooth surface functions.

The technique presented here is robust and requires no information concerning material geometry or flaw characteristics. Rather, it requires that the image comply with three assumptions: 1) image can be modeled as piecewise-smooth surface functions corrupted

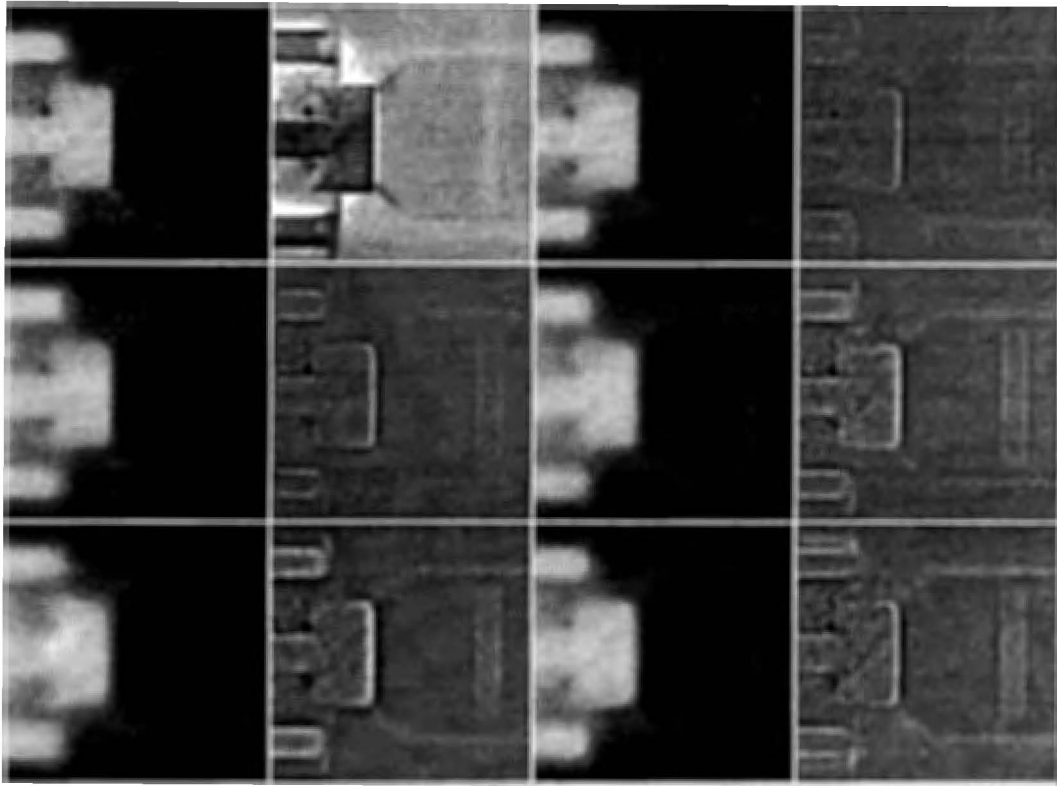


Figure 6.2: Results from existing techniques used to reduce variations due to geometrical structures

by noise and flaws, 2) large regions described by continuous smooth surface functions accurately approximate the image intensities due to geometrical structures within those regions, and 3) flaw areas are smaller than the smallest geometrical structure. Based on these assumptions, the approach involves segmenting the image into continuous surface regions via region growing and surface estimation techniques. Proper segmentation is achieved when regions representing relevant structure within the image are accurately identified. After the image is properly segmented, the unwanted features are reduced by subtracting the estimated intensities due to geometrical structures from the original image. The residual image is then assumed to contain intensity variations due to flaws and

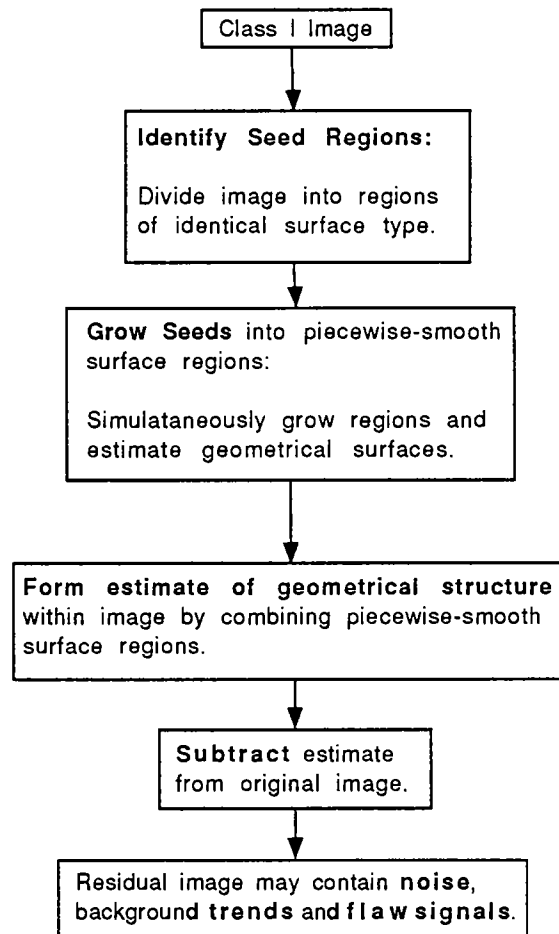


Figure 6.3: Flow diagram of four main steps in the approach to reduce the effects of geometrical structure

noise only. A flow diagram outlining specific steps performed in the removal process is shown in Figure 6.3.

In reference to the diagram, a Class I image is defined as an image that contains intensity variations due to geometrical structure. The remainder of the flow diagram will be described in the following section.

This approach is adapted from the algorithm developed by Besl and Jain [2] and

similar to the technique described by Taubin in [42]. Their algorithm is robust in properly segmenting images in a variety of applications including radiography. The segmentation algorithm is based on the assumption that the image data exhibits surface coherence in the sense that the data may be interpreted as noisy samples of a piecewise-smooth surface function. This assumption is consistent with the data acquired in the radiographic image formation process.

6.4 Description of Geometrical Structure Reduction Technique

Successful reduction of the variations resulting from geometric structures can only be achieved through accurate segmentation and approximation of the intensities in the digitized radiograph. Therefore, segmenting requires intelligent image interpretation procedures capable of successfully executing each processing step. The first step, as shown in Figure 6.3, involves identification of seed regions which are locally concentrated regions representative of the geometrical structures. Armed with seed regions, the second step attempts to grow the seeds into several large adjoining regions containing surface functions representative of the image intensities due to geometrical structures. The third step combines the individually estimated surface functions into a piecewise-smooth surface function approximating the intensities due to geometrical structures throughout the entire image. The final step involves subtracting the approximated surface intensities from the original image to produce a residual image that is void of these unwanted intensity variations.

To gain a better understanding of this procedure, we will discuss each one of the individual steps more thoroughly as they are applied to an actual digitized radiograph. A region of interest has been selected for testing and is outlined by the white box that

has been superimposed on the simulated radiographic image found in Figure 6.4. This simulated radiographic image shows the expected image intensities due to the specimen's geometrical structure. It should be noted that the reduction technique will be illustrated using digitized data from the selected region in an actual radiographic image. The selected region in the actual radiograph includes a ridge of thicker material that contains some porosity. The remainder of this section qualitatively discusses the steps shown in Figure 6.3. In Section 6.5, the steps are again discussed, but this time the explanation is quantitative.

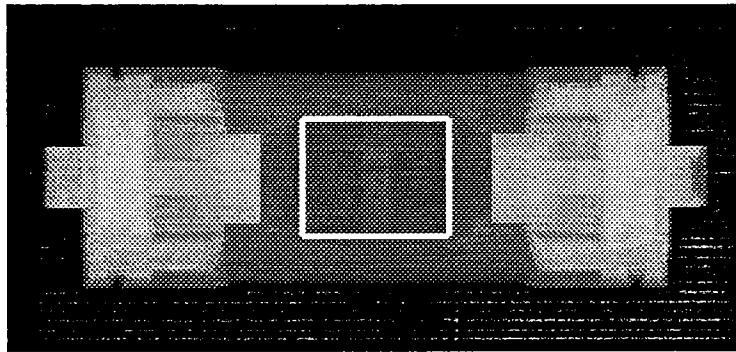


Figure 6.4: Simulated radiographic image of air conditioner part showing region of interest

6.4.1 Identifying seed regions

A difficulty encountered in all region growing based segmentation algorithms is that of selecting initial regions or seeds. Accuracy of the results and time involved are very dependent on the starting points selected. Due to the surface based nature of the algorithm, the seed regions are identified by finding locally concentrated regions possessing identical topological surface type characteristics. This is done by classifying each pixel in the image as belonging to one of eight fundamental surface types. The

classification is performed by analyzing the horizontal and vertical gradient information surrounding each pixel as described by Besl and Jain in [2]. Figure 6.5 shows the digitized region of interest and the result of classifying the image into the eight fundamental surface types represented by different grey levels. After classification, candidate seed regions are found by grouping pixels in the classified image with similar surface characteristics or grey levels. The final seeds are found by eroding the regions and selecting the larger remaining regions to guarantee the seeds are the interiors of the relevant structure. Figure 6.6 shows the four main seed regions identified in the test image.

6.4.2 Growing regions

In this region growing application, we want to approximate the pixel data in a region by a surface function that represents the relevant structure in an image. After seed

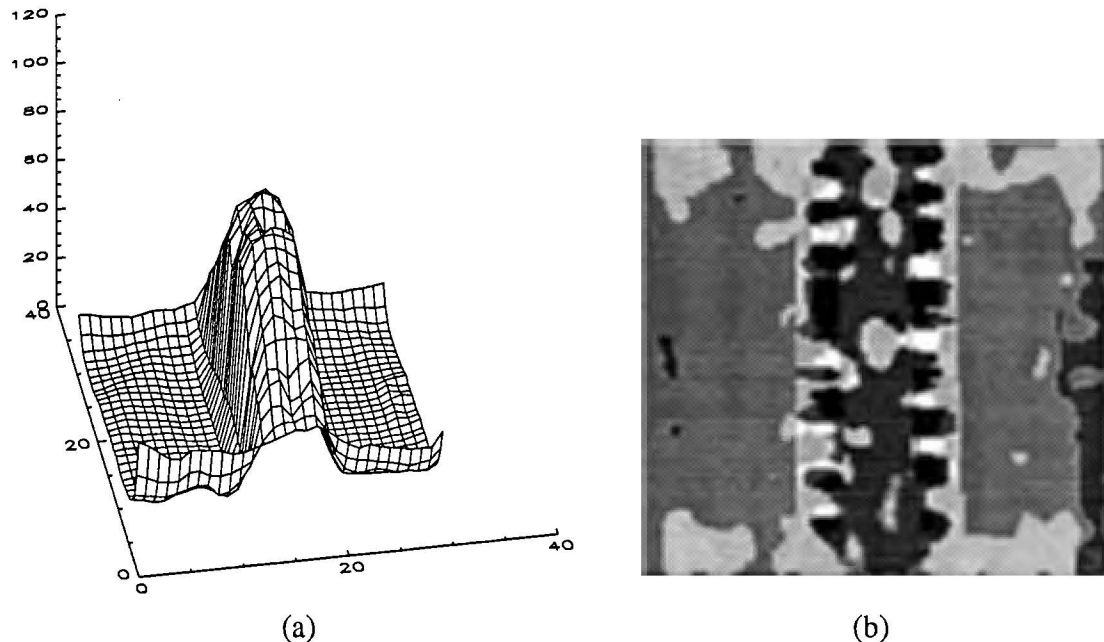


Figure 6.5: Results from classifying pixels into fundamental surface types: (a) surface plot of intensities in region of interest; (b) pixel classification results

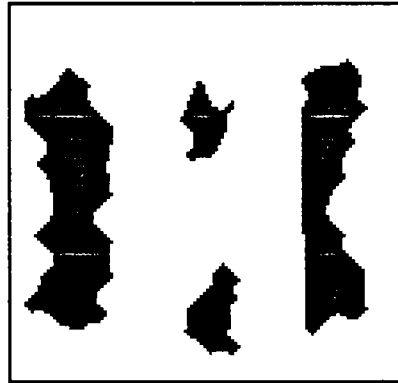


Figure 6.6: Image showing the four main seed regions identified.

regions have been identified, adjacent pixels are examined one by one and tested to see if they possess properties similar to those of the current region. The similarity property measured here is based on how well the surface function in the current region approximates the adjacent pixels. Adjacent pixels are accepted into the region if they are well approximated, and rejected if not. The region stops growing when no adjacent pixels are added. Additional growth constraints were introduced to limit the amount of uncertainty experienced in the model.

One of the growth constraints used by Besl and Jain involves a surface normal similarity measure. This similarity measure is found by comparing the angle between the approximated surface normal and the estimated surface normal. The approximated surface normal is determined by the polynomial coefficients for the approximating surface function and the image coordinates. The estimated surface normal is calculated using the horizontal and vertical gradient information generated by the seed identification algorithms. If the difference between the normals is too large then the candidate pixel is not added to the region.

Another growth constraint involves monitoring the shape of the growing region. This

constraint was introduced to encourage the seeds to grow into well defined continuous surface patches rather than narrow paths or a sparsely connected patterns of points. To eliminate these undesirable growth patterns, a topological descriptor was introduced into the region growing process. The descriptor is a function of the perimeter and the area of the region [1]. It is invariant to scale, translation, and rotation and therefore very useful in monitoring changes in the shape of the growing regions. The descriptor value V is calculated by dividing the area of a region N by the perimeter squared of that region P^2 and multiplying by 1000 for easy of handling where

$$V = 1000 * N/P^2. \quad (6.1)$$

Some sample shapes and descriptor values are shown below in Figure 6.7. From the examples, we see that large values of the descriptor are preferred to avoid narrow paths that fail to accurately segment the image.

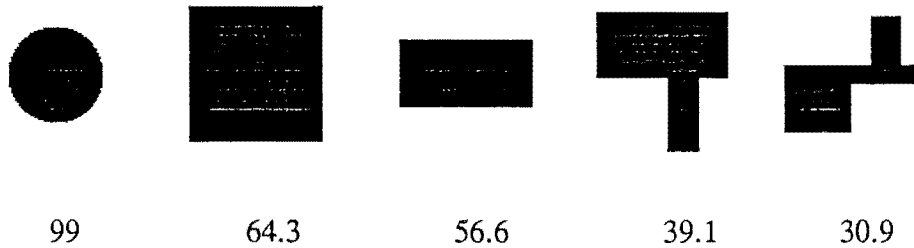


Figure 6.7: Sample region shapes and corresponding topological descriptor values

Based on this constraint, the two middle seeds in Figure 6.6 grew into a region possessing the surface function shown in Figure 6.8a. This approximated surface function can be compared with the actual image surface intensities shown in Figure 6.8b. The growth process continues for the remaining seed regions.

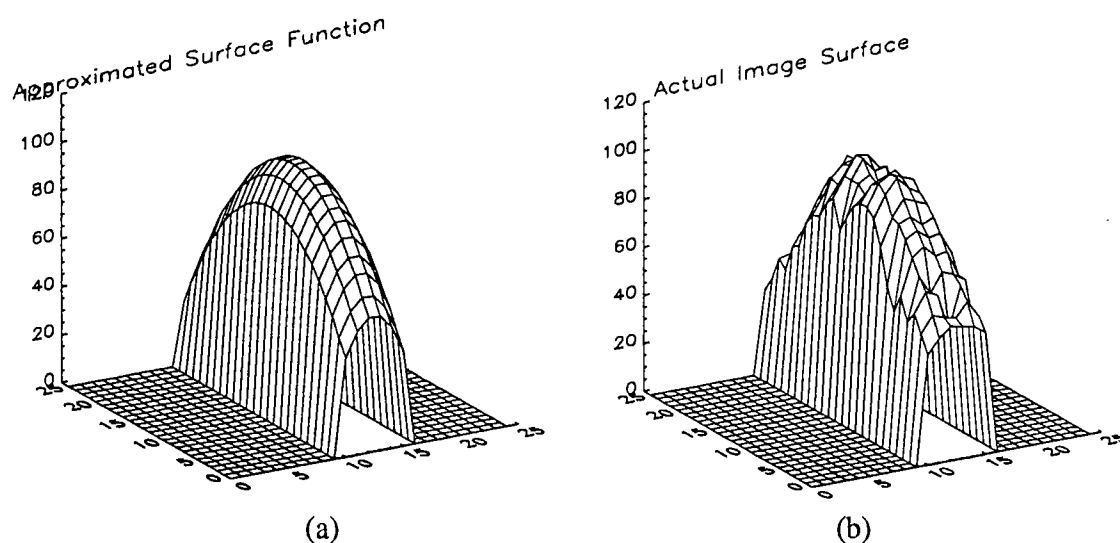
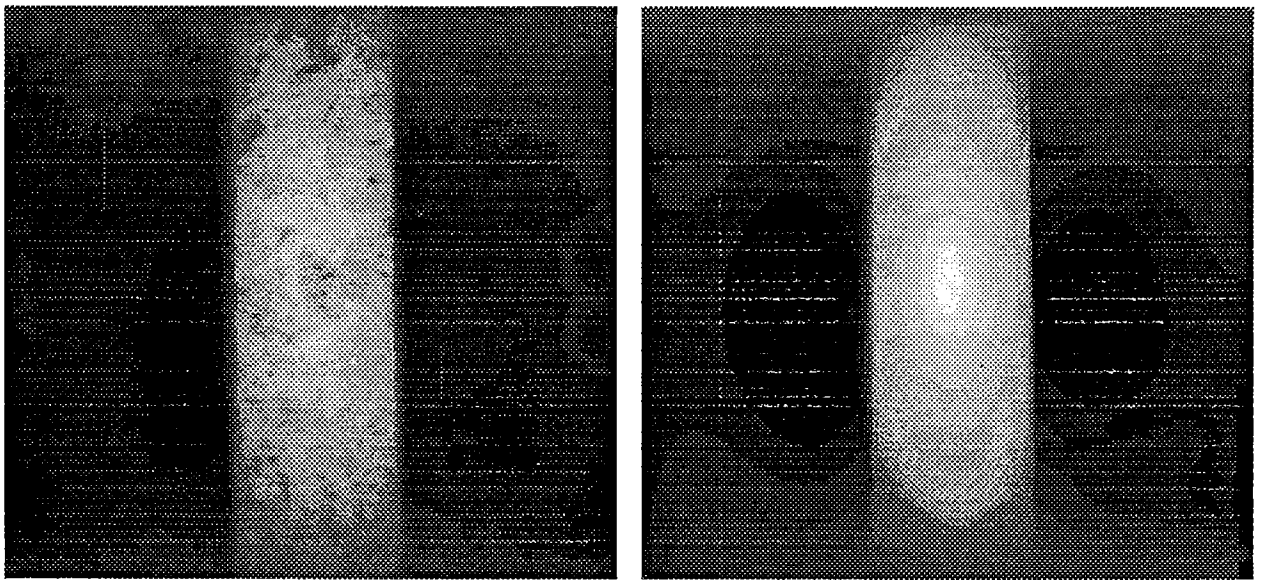


Figure 6.8: Surface plot of intensities in grown region: (a) least squares approximation of actual image surface in (b)

6.4.3 Merging grown regions

After the seeds have grown into large regions representative of the intensity variations due to geometrical structure, the corresponding surface functions are combined to form an estimate of the geometrical structure within the image. Often, there are gaps between the regions that were not accepted into any of the growing regions. These gaps may be areas experiencing abrupt transitions in image intensity due to geometric structure or noise. Larger gaps should be eroded, identified as seeds and sent back through the region growing process. Smaller gaps should be approximated by smoothly joining the region boundaries. Besl and Jain make reference to a region refinement technique and a one-step region merging method that can be found in [17]. A simple refinement technique has been developed for our application and is discussed in Section 6.5.3. Figure 6.9 shows the original region of interest and the estimated image found by merging the

piecewise-smooth surface functions using this technique. The image in Figure 6.9b gives no quantitative information concerning the accuracy of the VOSS technique in approximating the intensity variations due to geometrical structure. Rather, a quantitative assessment of the technique is performed after subtracting the approximated intensities from the original as discussed in the next subsection.



(a)

(b)

Figure 6.9: Comparison of original (a) and reconstructed images (b)

6.4.4 Subtracting the estimate

The final step in the geometrical structure reduction approach involves subtracting the estimated image, that represents intensity variations due to geometrical structures, from the original image. When the region growing based image segmentation algorithm has been successfully executed, the residual image will contain minimal intensity variations due to geometrical structure, but may possess features such as noise and flaw signals.

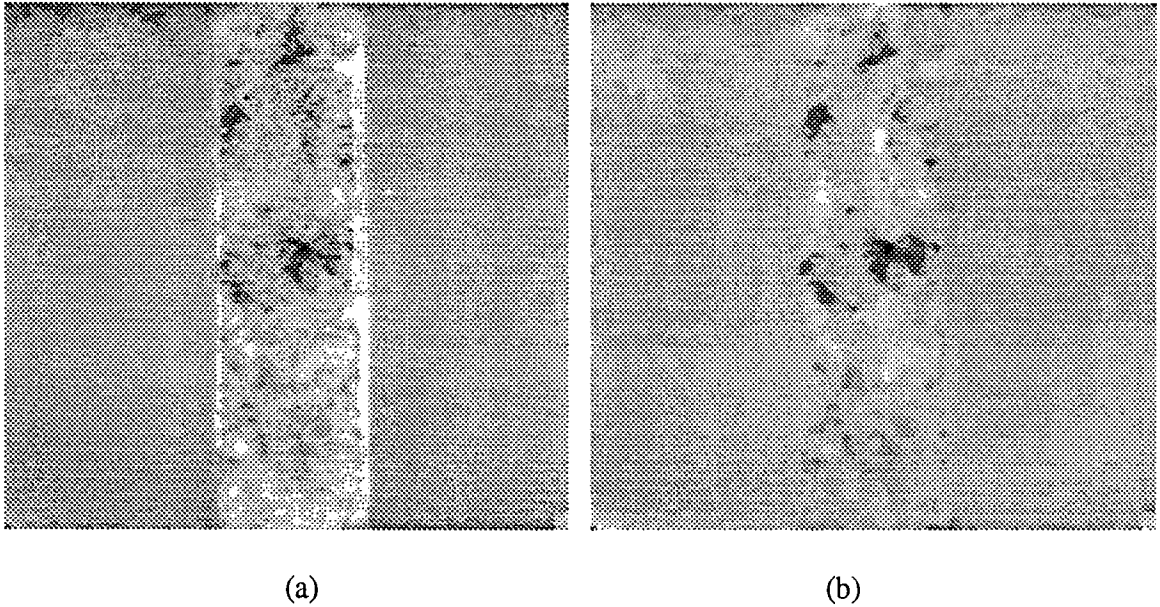


Figure 6.10: Geometrical reduction results: (a) residual image; (b) noise processed image

The residual image produced by processing the test image is shown in Figure 6.10a. Low contrast artifacts introduced by the sharp transitions in the original image can be seen on either side of the ridge of thicker material. The high contrast flaws, represented by the darker intensities, appear near the top and middle of the residual image and are very pronounced. Further processing in the Specific Processing stage (discussed in Chapter 7) will provide additional enhancement such as that shown in Figure 6.10b. A quantitative assessment of the technique is performed in the Specific Processing stage where the residual image will be thresholded to obtain the location and size of the candidate flaws.

6.5 Algorithm Description and Implementation

The technique described above is relatively simple in concept. The task of identifying the boundaries of the regions that outline the geometrical structures is not so difficult for

the human vision system. Rather, it is a trivial task for someone, trained or otherwise, to recognize the different patches or regions within the image that describe individual structures. The region of interest in the air conditioner part, for example, is easily partitioned by the eye into three individual regions; the lower intensity region on the left, the higher intensity ridge in the middle and the lower intensity region on the right. No matter how simple this task may seem, it is far from trivial when attempting to develop a computer algorithm to perform these operations using the input image data alone.

The subsections below describe the algorithms used to perform the different steps in the variable-order surface segmentation (VOSS) technique. The technique was implemented in the data analysis package called PV-Wave [21].

The main driver program, called *strt_itr(in_img, sd_lmt)*, is listed in Appendix B. This program initializes the necessary variables and provides the programs discussed below with the necessary inputs. The program also records the parameters of the segmented regions in a file for later reference.

6.5.1 Identifying seed regions

Seed regions are identified in the follow way. First, a surface type label image is generated from the original image. Appendix C shows the program listing called *surf_type(in_img, gu, gv)* that performs this function. Within this program, each pixel is classified as belonging to a particular surface type. This is done using partial derivative images (g_u , g_v , g_{uu} , g_{vv} , g_{uv}) which are calculated by convolving a smoothed version of the original image with convolution masks. The original image is smoothed to reduce the adverse effects of noise in determining the surface type. The first partial derivative images, $g_u(i,j)$ and $g_v(i,j)$, are calculated by one pass of the horizontal and vertical convolutions masks, respectively. The second partial derivative images, $g_{uu}(i,j)$, $g_{vv}(i,j)$,

and $g_{uv}(i,j)$, are found by a second pass of the correspond masks. Besl and Jain [2] suggest using a combination of the Mean and Gaussian curvature to obtain surface features that are invariant to rotations and translations within the image. The Mean curvature (H) and the Gaussian curvature (K) are found using the partial derivative images calculated above where

$$H(i,j) = \frac{(1 + \hat{g}_v^2(i,j))\hat{g}_{uu}(i,j) + (1 + \hat{g}_u^2(i,j))\hat{g}_{vv}(i,j) - 2\hat{g}_u(i,j)\hat{g}_v(i,j)\hat{g}_{uv}(i,j)}{2(1 + \hat{g}_u^2(i,j) + \hat{g}_v^2(i,j))^{3/2}} \quad (6.2)$$

and

$$K(i,j) = \frac{\hat{g}_{uu}(i,j)\hat{g}_{vv}(i,j) - 2\hat{g}_{uv}(i,j)}{(1 + \hat{g}_u^2(i,j) + \hat{g}_v^2(i,j))^2} \quad (6.3)$$

Eight fundamental surface types can be identified using only the sign of the Mean and Gaussian curvature. This is shown in Table 6.1. Figure 6.11 shows the eight fundamental surface types referred to in this table.

The signs of the Mean and Gaussian curvature values at each pixel location are used with the table above to label each pixel with the corresponding surface type value T.

Table 6.1. Surface type labels and values from H and K curvature signs

	K > 0	K = 0	K < 0
H < 0	Peak T=1	Ridge T=2	Saddle Ridge T=3
H = 0	none T=4	Flat T=5	Minimal Surface T=6
H > 0	Pit T=7	Valley T=8	Saddle Valley T=9

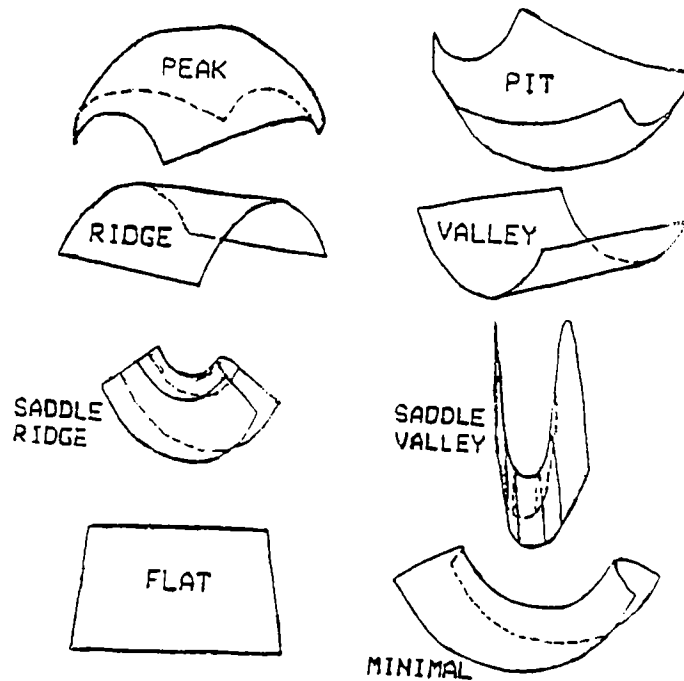


Figure 6.11: Eight fundamental surface types used to classify each pixel

After a surface type label image $T(i,j)$ has been generated, it must be processed to isolate interior seed regions possessing the same topological surface types. These seed regions are representative of the intensity variations due to geometrical structures. Appendix D lists the program that identifies the interior seed regions.

The task of identifying interior seed regions begins in procedure *seed_rgns(in_img, timg, sd_lmt, gm_lmt)*. First, the surface type label image $T(i,j)$ is processed by two passes of a 5x5 median filter. This eliminates small islands that lay within the boundaries of a larger region. These small regions originate from noise and flaws in the original image. This is followed by an erosion operation that ensures neighboring seed regions are not touching one another. The next step involves identifying all the individual regions

that contain more than sd_lmt pixels, where sd_lmt is the minimum number of pixels required to represent a seed. The candidate seed regions are then sorted according to size. This information is used later in the region growing procedure. At this point Besl and Jain suggested eroding each candidate seed region until it was less than or equal to the minimum seed size. This method was not found to be reliable in providing the best candidate seed regions for our application. Instead, an iterative method was developed to find the best candidate seed regions.

The iterative method used here is designed to identify the smallest seed regions possessing the most favorable shape characteristics. For this reason, the topological descriptor mentioned in Section 6.4.2 was implemented here to help characterize the seed's shape. The threshold value gm_lmt was defined as the minimum allowable topological descriptor value for a seed region. The algorithm iteratively erodes each candidate seed region. Between successive erosions, the algorithm finds the largest remaining region $lrgn$ and processes it with the procedure $xpnd(lrgn, rgn)$. The $xpnd()$ function grows $lrgn$ out to the boundary of the original seed until its topological descriptor decreases, $xpnd$ then returns the grown region to the iterative loop. The algorithm compares the candidate seed's size and topological descriptor value with sd_lmt and gm_lmt throughout the iterative procedure and quits as soon as the seed contains fewer than sd_lmt pixels or achieves a topological descriptor value greater than gm_lmt . This iterative procedure is performed on each candidate seed region and provides the region growing procedure with reliable seed regions.

6.5.2 Growing regions

Region growing is a crucial component of the VOSS technique. The regions must grow out to the boundaries of the geometrical structure while staying insensitive to

noise and fluctuations in flaw signals. The algorithm must know when the regions should stop growing and when higher order bivariate polynomials should be used. The algorithm must also determine when to increase the allowable fitting error threshold for adding additional pixels. All of these factors are implemented into the region growing procedures. Appendix E list the code used for this portion of the VOSS technique.

The region growing algorithm *grower(in_img, c_rgn, coef, rgn_parm)* is the central control program for the region growing procedure. This program will iteratively call the function *rgn_grwr()* which grows the region specified by *c_rgn* into a larger region. This larger region, called *new_rgn*, is made up of *c_rgn* and adjacent pixels that are compatible with the surface function approximating the original image intensities. An adjacent pixel is defined as compatible when the magnitude of the difference between the observed intensity and the approximated intensity is less than the current allowable fitting error *ns_lmt* and the pixel satisfies the surface normal constraint. A check for compliance with surface normal constraint is performed in the procedure *nrml()* as outlined in Besl and Jain [2]. An additional constraint was applied to the region growing algorithm to improve the segmentation results for NDE X-ray applications. This constraint involves monitoring the growth of the region to determine when the region is not growing in a well defined manner as described in Section 6.4.2.

The algorithm monitors the region's growth between successive calls to *rgn_grwr()* by measuring two of the region's characteristics; size and topological descriptor value. As the region grows, the change in size and the change in the topological descriptor value are monitored to help the algorithm establish when the region's growth has slowed or its shape has started to developed peculiar characteristics. Undesirable growth patterns are detected by a combination of significant decreases in the topological descriptor value

and small growth rates.

The percent change in size, rg , is calculated by dividing the number of new pixels added, new_pix , by the total number of pixels occupying the region, ar . Two separate measures are implemented to monitor the region's shape. The first, ctd , measures the change in the topological descriptor value and is calculated by dividing new_pix by the change in the perimeter squared. The second, rtd , measures the percent change in the topological descriptor value. This measure is calculated by subtracting the current value from the previous value and dividing by the previous value.

These measurements were plotted on a chart as they were calculated to provide a visual relationship between the measurements and the behavior of the growing region. This was extremely beneficial to the process of empirically selecting the appropriate thresholds to signal abnormal growth patterns and slowing growth rates. This threshold selection process has undergone several refinements. The current thresholds have been established after observing the growth of many different regions within several different images. The established decision rules for accepting a group of compatible pixels are shown in Figure 6.12.

The variable-order nature of this "data driven" technique enables it to adapt to the image data. The number of approximating functions required and their complexity are dependent on the types of variations experienced in the image data. The number of approximating functions is limited to three in our application.

The functions used here include planar, biquadratic, and bicubic functions where

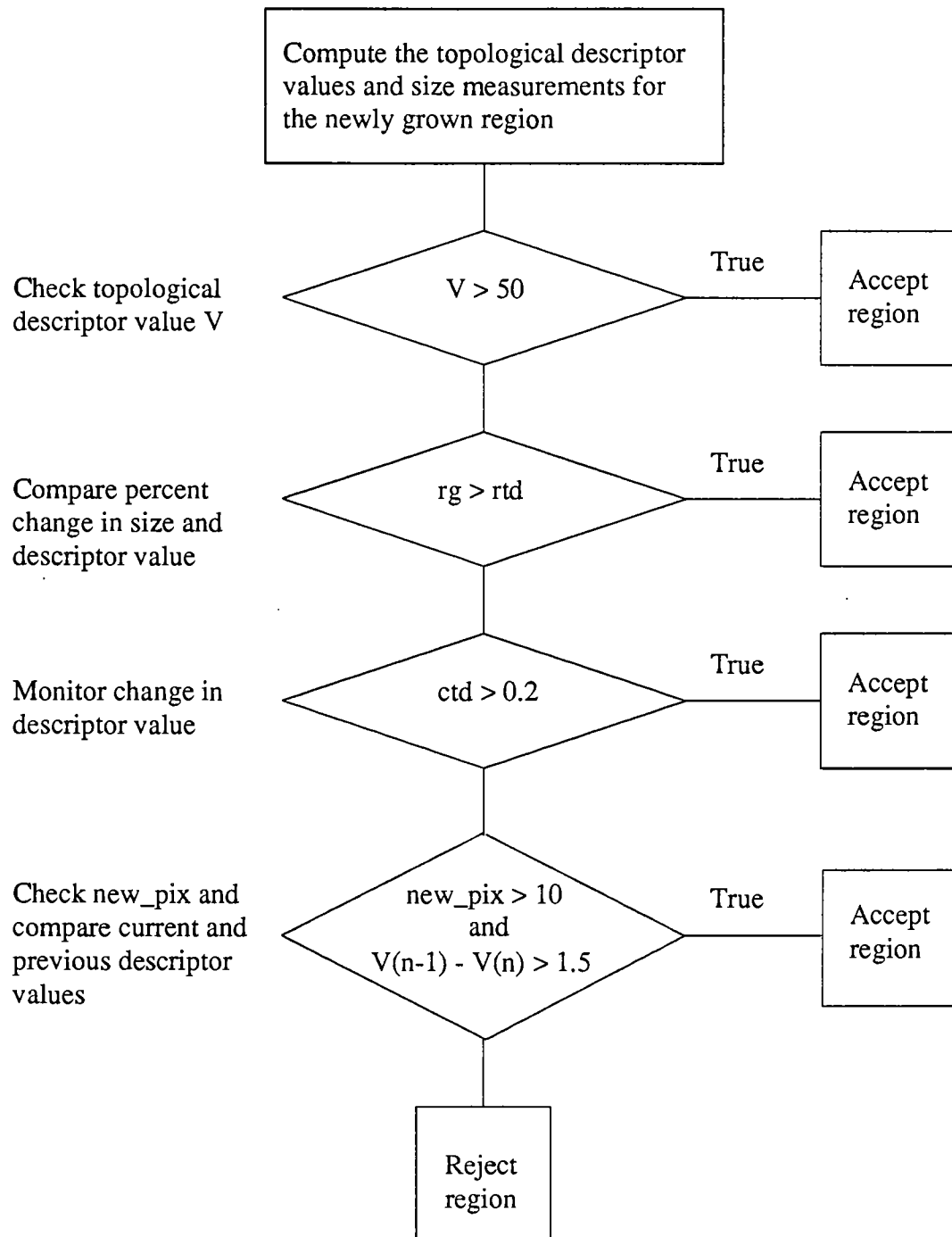


Figure 6.12: Flow diagram showing decision rules for accepting newly grown region

$$\begin{aligned}
f_{x,y} = & a_{00} + a_{10}x + a_{01}y \\
& + a_{11}xy + a_{20}x^2 + a_{02}y^2 \\
& + a_{21}x^2y + a_{12}xy^2 + a_{30}x^3 + a_{03}y^3.
\end{aligned} \tag{6.4}$$

The programs that prepare the image data for the least squares fitting and approximating procedures are called *crt_mtrx()* and *rgn_appx()*, respectively. The program that performs the least squares fitting is called *lssfit()*. These programs are listed in Appendix F.

When no additional pixels are found to be compatible with the function describing the intensities in the current region, the algorithm selects the next higher order bivariate polynomial and sends the current region back into the region growing procedure. The region terminates the growing process when the next higher order function fails to achieve a region much larger than the previous or the highest order function has been reached. This process continues for the remaining seed regions.

After a seed region is grown, the approximated surface function is subtracted from the original image to obtain the fitting error for each pixel location within that region. A residual image, called *err_img*, is maintained for the succeeding growth processes. As a new region is grown and overlaps a previous region, this image is used to compare previous fitting errors with the fitting errors incurred by the new region. Overlapping pixels will be accepted into the new region if they provide a better fit than that recorded in *err_img*. This allows the algorithm to correct any erroneous region boundaries created earlier.

6.5.3 Merging Grown regions

Following the termination of the last region, the algorithm checks to see if all the

pixel locations have been accepted into one of the grown regions. Most often, there are gaps between some of the regions. These gaps may represent areas experiencing abrupt transitions in image intensity or areas possessing noise and flaw signals that were not compatible with the adjacent regions. Larger gaps are identified as seed regions in the program *next_rgn()* and sent back through the region growing process. This program is listed in Appendix G.

After growing the larger gaps, the remaining gaps can be dealt with using the region refinement approach suggested by Besl and Jain [17] or the simple approach developed here. The approach discussed here associates the pixels in the gap with one of the adjacent regions. The algorithm that performs this operation can be found in Appendix H.

The name of the program is *fill_holes(f_rgns, in_img, f_coef, hls)*. The program fills the pixels in each gap with the neighboring approximating function that best matches the original image data. This is done in an efficient and speedy manner that yields useful results in most situations.

6.5.4 Subtracting the estimate

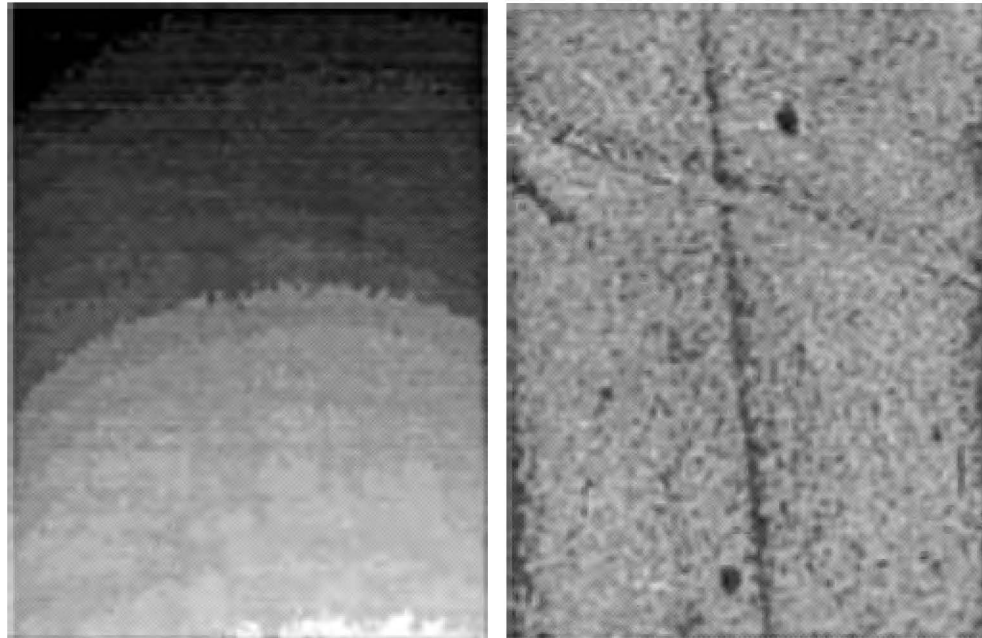
After the approximated image has been reconstructed by merging the grown regions together, the algorithm attempts to reduce the effects of geometrical structure by subtracting the reconstructed image from the original. Some results obtained using the VOSS technique to reduce the intensity variations due to geometrical structures as show in the following section.

6.6 Geometrical Structure Reduction Results

This VOSS technique has been tested on several radiographs possessing a variety of unique intensity variations due to the geometrical structures of the specimen. Some of

these examples are shown below.

Figure 6.13a shows a portion of a radiograph with a large dynamic range of film densities. The feature of interest, buried in the image background, is a long thin crack that extends diagonally from top to bottom in the image. The noise variance within the image is 3.7 and the flaw signal is approximately 4 grey levels off the average background signal. This is illustrated in Figure 6.14 which shows a plot of two horizontal slices through the middle of the original image. The thicker line is the average of six consecutive rows. The flaw signal can be found in this plot between horizontal positions 50 and 55 on the x-axis. The thinner line is offset from the previous and shows the data found in one of the six consecutive rows. Figure 6.13b shows the result of reducing the geometrical background structure using the VOSS technique developed here. Several different features of interest



(a)

(b)

Figure 6.13: Geometrical reduction results: (a) original image; (b) residual image

are observed in this image. The flaw that extends diagonally from top to bottom is more easily detected. On either side of the flaw, there are small circular features introduced by foreign objects such as dust or water spots lying on the lens or Vidicon imaging device. The presence of these features stresses the importance of maintaining the image capturing equipment. A third feature noticed through visual inspection is the diagonal line in the upper half of the image. This feature contains a smooth background that is clearly different than the noisy background observed elsewhere. This is due to the inherent noise characteristics of the original image data within this transitional area.

Figure 6.15 shows slices through the residual image corresponding to the same row positions plotted in the previous figure. The technique approximated the variations due to geometrical structure with piecewise-smooth surface functions that are not affected by the small variations due to the flaw signal. Because of this, the geometrical structure reduction technique demonstrated the ability to reduce the intensity variations due to geometrical structures without introducing significant artifacts or altering the flaw signal. This was due in part to the absence of abrupt transitions due to geometrical structures. The variations due to flaws and structure varied less than 20 intensity level over 5 pixels. After additional feature extraction processing in the Specific Processing stage, the residual image will be thresholded and candidate flaws will be identified. Figure 6.16 shows the results of thresholding the processed image to find the candidate flaw locations. Although the flaw signal was only 4 grey level off of the noisy background signal, the VOSS processing technique adequately preserved the flaw signal so that the Specific Processing stage could find the flaw using a simple thresholding technique as shown in Figure 6.16.

Figure 6.17 shows a region of interest from the air conditioner part used in Section

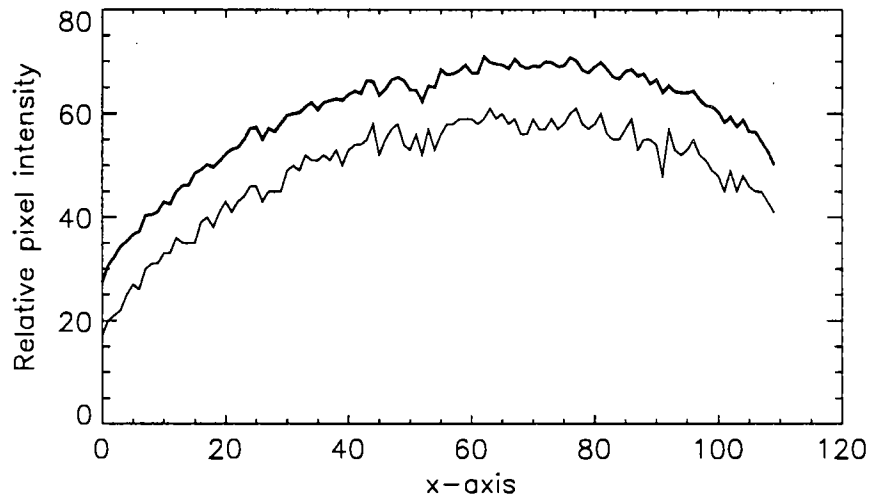


Figure 6.14: Plot of row data in original image

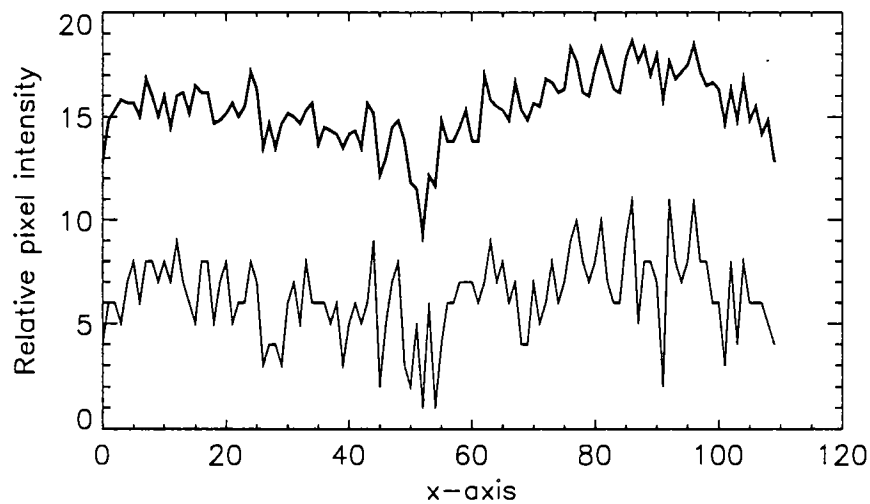


Figure 6.15: Plot of row data in residual image

6.4. Like the previous example, this image also possesses a large dynamic range of film densities. The image also contains some sharp transitions in intensity due to the specimens geometrical structure. The approximated intensities due to geometrical structures found using the VOSS technique are shown in Figure 6.18. This image seems to provide an accurate piecewise-smooth approximation of the intensity variations due to geometrical structures. However, the sharp intensity variations presented some problems for the

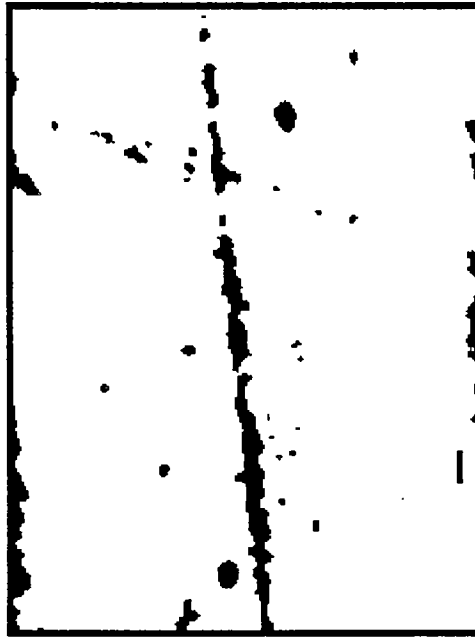


Figure 6.16: Thresholding results for image in Figure 6.13b

segmentation algorithm. This can be seen in the residual image shown in Figure 6.19 which was found by subtracting the approximated image from the original. The residual image contains some artifacts that were introduced by the subtraction process. These artifacts resulted from a poor approximation of the original image intensities in the neighborhood of the abrupt changes due to geometrical structures. Figure 6.20 show the quantitative results that would be analyzed by the computer in the Specific Processing stage. The various islands within this image would be identified as candidate flaws and more thoroughly analyzed by feature extraction and statistical decision making algorithms to determine their significance as discussed in Chapter 7.

The VOSS technique accurately approximated the original image intensities in transitional areas experiencing variations less than 35 grey levels over 5 pixels. The technique was not able to accurately approximate these areas when the variation was greater than 40

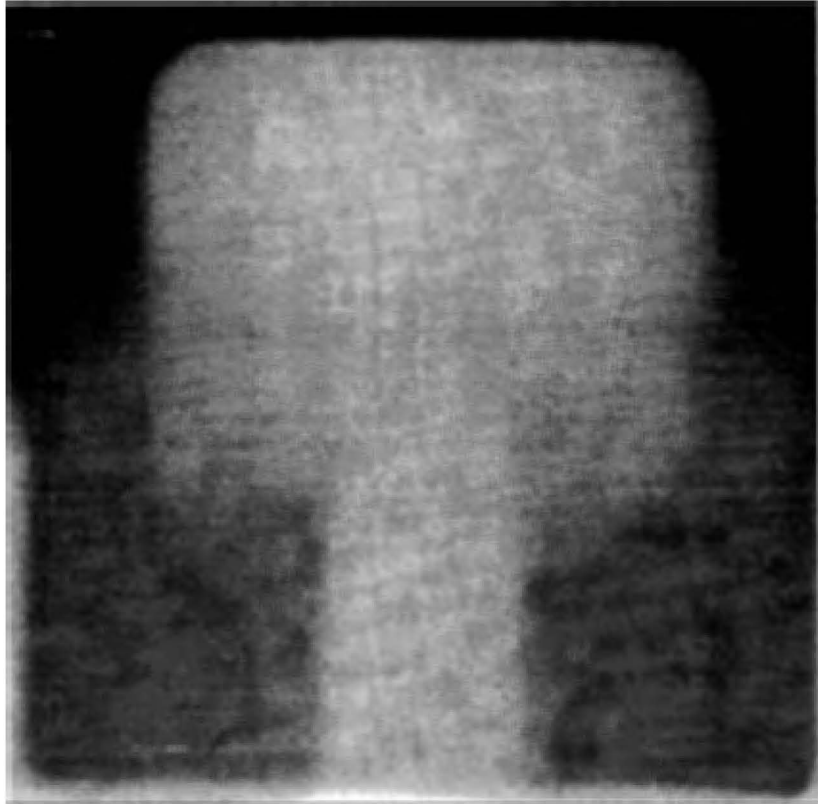


Figure 6.17: Portion of original radiographic image of air conditioner part

grey levels over 5 pixels. This is mainly due to the inability of the bivariate polynomial functions to approximate these sharp transitions between grown regions. In an automated flaw detection system, these poor approximations would be identified as candidate flaws. The Specific Processing stage would be required to distinguish between artifacts introduced by this technique and variations due to flaw signals. This could be achieved by providing the Specific Processing stage with information concerning the grown region boundaries and by analyzing the original image intensities in these areas as discussed in Chapter 7. The effect of these undesirable artifacts may be reduced by using more sophisticated region merging techniques. This would be the focus of future research

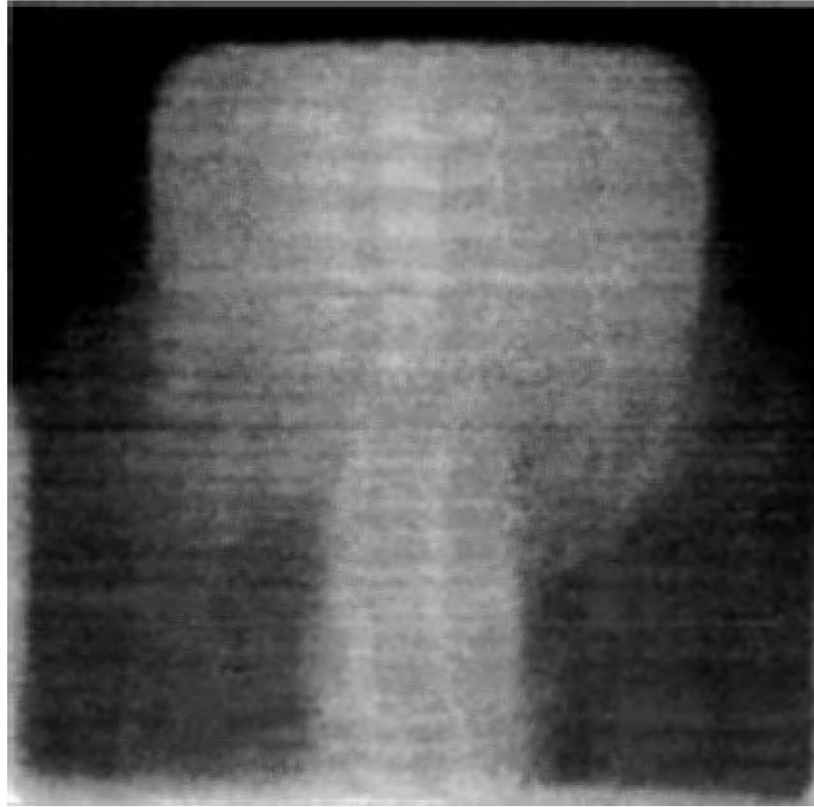


Figure 6.18: Approximated image intensities found using VOSS technique

concerning the VOSS technique and its application to reducing the effects of geometrical structures within radiographic images.

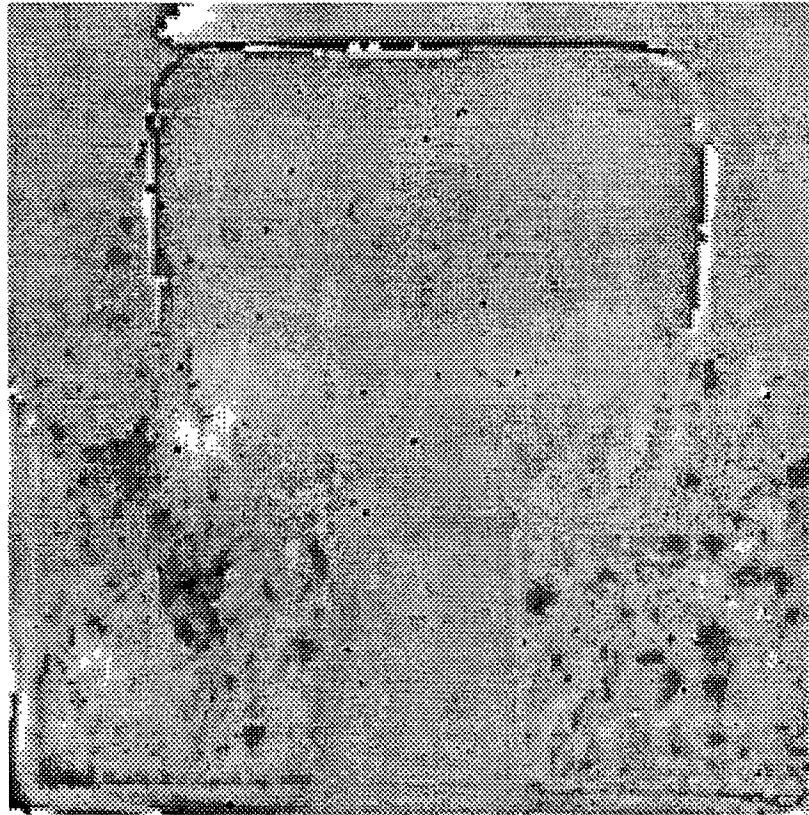


Figure 6.19: Residual image intensities found by subtracting approximated image from original

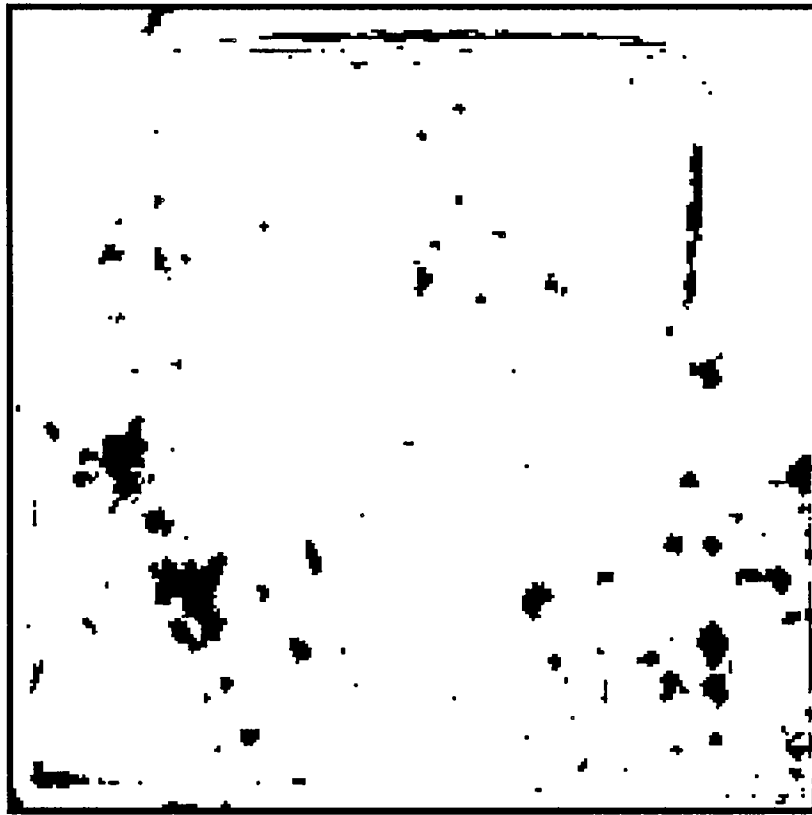


Figure 6.20: Candidate flaw locations found by thresholding the residual image in Figure 6.19

CHAPTER 7. SPECIFIC PROCESSING STAGE

The proposed automatic flaw detection scheme is divided into two separate stages as mentioned in Section 3.6 and shown in Figure 3.1. The General Processing stage includes the processes discussed in Chapters 4-6. These processes are used to reduce the contributions from unwanted features within the image. The success of the General Processing stage relies on the ability of these processes to accurately identify and remove the intensity variations resulting from the unwanted features without introducing artifacts or altering information concerning flaws in the image.

Based on this idea, the Specific Processing stage is designed to identify, classify and assign a confidence level to candidate flaws found in the processed image. The Specific Processing stage can be tailored to each particular application by incorporating information concerning flaw characteristics and other relevant information. The Specific Processing stage would also use statistical measurement techniques and apply detection and estimation theory principles to develop probability of detection and probability of false alarm models for flaws in the post-processed images. The visual quality of the processed image will not be factor in determining the acceptability of radiographed specimen, rather, the automated system will base its decision on quantitative information alone. Therefore, these quantitative measurements and models will play a major role in developing a successful automated flaw detection system. Some quantitative analysis methods used to numerically assess the qualitative results are discussed below.

7.1 Quantitative Assessment of Processed Image

Quantitative assessment of the processed image is the principle role of the Specific Processing stage. This section will focus on the quantitative measurements and processes that will interpret the features remaining in the processed image.

The first step in this interpretation process would be to enhance the suspect flaw features in the processed image. This would be performed by specialized feature extraction techniques that incorporate a priori information concerning the flaws' characteristics. Some of the feature extraction techniques discussed in Section 2.5.3 would be applied here. These techniques would most likely include matched filtering, mathematical morphology, and convolution operations.

After enhancing the processed image, automatic thresholding algorithms would be used to identify candidate flaw locations. These candidate flaw locations would provide an initial guess concerning the location of flaws within the image. Candidate flaw locations would then be more thoroughly scrutinized by applying statistical analysis algorithms that analyze the data in both the original and processed images. Statistical analysis techniques used to discriminate between variations due to flaws and variations due the material's characteristics would be performed here. This cross validation method of analyzing areas corresponding to candidate flaw locations in both the processed and original image would also be helpful in distinguishing between introduced artifacts and valid flaw features. These results could then be processed by artificial neural network based algorithms that have been trained to identify flaw characteristics common to each particular application.

Flaw sizing and classification would be the next step in numerically analyzing the information remaining in the processed images. Edge detection algorithms could be used here to identify the flaws boundaries. Then, existing flaw sizing algorithms would

be implemented. These sizing algorithms could be modified to utilize information gained by the feature identification processes in the General Processing stage. For example, the approximated background trend intensity at a specific location in the image could be converted to a film density value or part thickness value when the material absorption coefficients and X-ray generator settings are known. This material thickness value could then be used along with local intensity variation measurements to determine the actual thickness of the candidate flaw. Flaw classification procedures could also use this information to determine the severity of the candidate flaw.

Probability of detection models could also be developed to predict flaw detection and flaw characterization performance in the processed images. These models would be based on estimation and detection theory principles and would also utilize information gained by the feature identification processes in the General Processing stage. Criterion-of-goodness measurements should also be developed and implemented in these models to generate quantitative measures of the scheme's performance for each particular radiograph inspected. Furthermore, these statistical measurements will be a significant factor in assigning confidence levels to the candidate flaws.

The Specific Processing stage will make the final decision concerning the acceptability of the inspected radiograph in a fully automated flaw detection system. Therefore, careful consideration must be given to determining the accuracy and reliability of the processes and decision making procedures involved. Future work in this area should focus on the investigation and development of statistical measurement models that will complement the feature identification and reduction processes in the General Processing stage.

CHAPTER 8. SUMMARY AND CONCLUSIONS

A general automatic flaw detection scheme has been developed to automatically reduce the effects of unwanted image features and thereby prepare an arbitrary digitized NDE radiographic image for flaw extraction, sizing and classification algorithms. This scheme is based on the idea of removing the undesirable image features so that all that remains is the pertinent information such as candidate flaw signals. The scheme has been divided into two separate stages. The first stage, General Processing, performs the processes deemed necessary in reducing the contributions from unwanted features within the image. These are undesirable features common to radiographic images that tend to confuse flaw extraction and identification algorithms. The later stage, Specific Processing, is tailored for each application and performs flaw detection and classification procedures by incorporating information concerning flaw characteristics relevant to that particular application. This stage involves processes that will quantitatively assess the information remaining in the processed image to determine the specimens acceptability. This thesis focuses on the automated feature reduction processes in the General Processing stage.

8.1 Analysis of Processing Results

Existing image processing techniques were modified and new techniques were developed to achieve accurate and reliable feature reduction results. In particular, the modified adaptive smoothing filter was developed to automatically reduce the effects of noise within the image without influencing the existing flaw signals. Existing nonparametric

noise filtering techniques tested on void and crack type flaws produced inferior results. In addition to identifying successful noise filtering techniques, normalized noise measures were developed to quantitatively characterize the images noise content. These measurements were then used to determine the appropriate techniques required to suppress the noise in an arbitrary radiographic image. Future work in the area of automated noise reduction would benefit from the development of additional noise measurement tools capable of reliable noise quantification in images with complex background structures.

Current trend removal techniques, based on a fixed background trend model, were sufficient in reducing the effects of background trends. Normalized image measurements were developed to identify the trend and determine the appropriate model parameters for reducing it. This technique was successful in removing 90% of the unwanted intensity variations within the noisy Martin Marietta weld image in Figure 5.2 by performing automated trend reduction. The remaining unwanted intensity variations were mainly due to noise. The image of the pipe weld in Figure 5.9 had 88% of the unwanted intensity variations removed using this technique. The resulting images contained much smaller ranges in intensities by removing the unwanted background trends. This allows the flaw detection algorithms to better discriminate between flaw signals and other image features. Both of the examples yielded favorable results since the trends were adequately described using the one-dimensional polynomial model. Automated trend reduction performs well in many situations. Future work in this area should focus on trend reduction techniques that adapted to the observed background trends which are not adequately described by the fixed polynomial model.

Due to the lack of existing techniques, a totally new approach was pursued in developing a robust technique to automatically identify and reduce the intensity variations

due to the specimen's geometrical structure. The approach involves segmenting the image into piecewise-smooth surface functions that approximate the intensity variations due to geometrical structure. Although the variable-order surface segmentation concept involved is relative simple, the modelling uncertainty associated with non-ideal data experienced in NDE radiography presents a challenging problem. This technique was implemented and additional algorithms and measurements were developed to improve its performance in NDE X-ray applications. The technique performed relatively well on images containing less complicated geometrical structures. However, images possessing extremely sharp intensity variations presented some problems since the abrupt changes in intensity were not easily forgiven by the bivariate polynomial models.

The automated geometrical structure reduction technique performed well on several examples as shown in Chapter 6. The results can be qualitatively measured by calculating the percent of unwanted intensity variations removed by the technique. For the selected region of the air conditioner part shown in Figure 6.17, it was found that 98% of the unwanted intensity variations due to geometrical structure were removed. This is confirmed by a change in intensity range of 140 grey levels for the original to a range of 20 for the residual. The flaw sizes for this image ranged from 10x10 to 10x40. Because of its size, the largest flaw was identified as an intensity variation due to geometrical structure. This resulted in altering the flaw signal. This can be avoided by selecting a larger threshold for the smallest allowable geometrical structure. Abrupt transitions within this image were not accurately approximated and therefore the residual image contained artifacts with intensities varying from 10 to -10 as shown in Figure 6.19. Future work in this area should focus on the development of additional region merging techniques that adapt to these abrupt changes. Subsequent noise processing may also help to reduce the

affects of these artifacts, but it will not eliminate them. More powerful discrimination techniques that consider the size and shape of these artifacts may also be necessary in discriminating between true flaws and introduced artifacts as discussed in Section 7.1.

Another image shown in Figure 6.13 was successfully processed by the automated geometrical structure reduction technique. The technique was able to remove 99% of the intensity variations due to geometrical structures. The range of the original image was 230 grey levels and the processed image possessed only 24 grey levels. This significant reduction in unwanted intensity variations prepared the image for flaw detection and sizing techniques that could quantitatively assess the radiographic image for flaws. The width of the crack-like flaw in this particular image was only 5 pixels across and was not much stronger than the surrounding noise. This type of situation requires sensitive processing techniques that do not alter the flaws' characteristics.

The processing techniques performed by the General Processing Stage were found to be very effecting in reducing the unwanted intensity variations and preparing the arbitrary radiographic images for automatic flaw detection processing. The processed images were quantitatively assessed by applying simple automatic thresholding algorithms that provide a preliminary discrimination between flaws and other image features. The threshold examples found in Chapters 4-6 illustrate the effectiveness of the noise, trend, and geometric structure reduction subprocesses in sufficiently reducing the unwanted intensity variations so that the Specific Processing stage can accurately identify and size the candidate flaws.

The processed image produced by the subprocesses in the General Processing scheme must be quantitatively analyzed by the automatic flaw detection system. This analysis would be performed in the Specific Processing stage by algorithms based on statistical

estimation theory principles and models. These algorithms would be capable of incorporate a priori information concerning the expected flaw characteristics for each particular application. Probability of detection models would also be an important part of the quantitative assessment and would utilized information gained by the feature identification processes in the General Processing stage. In addition to the sizing and classification information calculated for the candidate flaws, the system would use various statistical measurement models to assign a confidence level to each candidate flaw. Future work in this area should focus on the investigation and development of statistical measurement models that would complement the feature identification and reduction processes in the General Processing stage. This work would be an integral part of the automated flaw detection scheme and facilitate the integration of the scheme into reliable automatic flaw detection systems.

The general automatic flaw detection scheme developed here demonstrates the ability of digital image processing techniques to automatically identify candidate flaws in arbitrary radiographic images. This scheme is capable or reducing the effects of a variety of confusing image features that inhibit traditional thresholding methods from accurately discriminating between a flaw and other image features. This scheme accommodates a variety of NDE radiographic inspection applications by providing for the integration of a priori knowledge concerning flaw characteristics and other image features in the Specific Processing stage. This thesis demonstrates the feasibility of developing reliable automatic flaw detection systems for a variety of NDE applications.

REFERENCES

- [1] D.H. Ballard and C.H. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [2] P.J. Besl and R.C. Jain. Segmentation through variable-order surface fitting. *IEEE Trans. PAMI*, 10(2):167–192, March 1988.
- [3] Herbert Boerner and Helmut Strecker. Automated X-ray inspection of aluminum castings. *IEEE Trans. PAMI*, 10(1):79–91, January 1988.
- [4] A.C. Bovik. Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Trans. Signal Processing*, 39(9):2025–2043, September 1991.
- [5] Richard A. Brown. Image processing workstation software development and feature size measurement methods for NDE X-ray images. Master's thesis, Iowa State University, Ames, IA, 1992.
- [6] M.S. Chackalackal and J.P. Basart. NDE X-ray image analysis using mathematical morphology. In D.O. Thompson, editor, *Review of Progress in Quantitative Non-Destructive Evaluation*, volume 9a, pages 721–728. Plenum Press, New York, 1990.
- [7] P.C. Chen and T. Pavlidis. Segmentation by texture using a co-occurrence matrix and a split and merge algorithm. *Comput. Graphics Image Processing*, 10:172–182, 1979.
- [8] P.C. Chen and T. Pavlidis. Image segmentation as an estimation problem. *Comput. Graphics Image Processing*, 12:153–172, 1980.
- [9] E.R. Doering. Detection of Anomalies in Digital Images Using Pixel Classification. Master's thesis, Iowa State University, Ames, IA, 1987.
- [10] E.R. Doering. *Three-dimensional flaw reconstruction using a real-time X-ray system*. PhD thesis, Iowa State University, Ames, IA, 1992.

- [11] E.R. Doering and J.P. Basart. Trend removal in X-ray images. In D.O. Thompson, editor, *Review of Progress in Quantitative Non-Destructive Evaluation*, volume 7a, pages 785–794. Plenum Press, New York, 1988.
- [12] T.J. Fan, G. Medioni, and R. Nevatia. Segmented descriptions of 3-d surfaces. *IEEE J. of Robotics and Automation*, 3(6):527–538, December 1987.
- [13] O.D. Faugeras, M. Hebert, and E. Pauchon. Segmentation of range data into planar and quadratic patches. In *Proc. IEEE-CS Conf. Comput. Vision Pattern Recognition*, pages 8–13, New York, NY, 1983. IEEE Press.
- [14] R.C. Gonzalez and P. Wintz. *Digital Image Processing 2nd Ed.* Addison-Wesley, Reading, MA, 1987.
- [15] R. Halmshaw. *Industrial Radiology: Theory and Practice*. Applied Science Publishers Ltd, England, 1982.
- [16] J. Han, R. Volz, and T. Mudge. Range image segmentation and surface parameter extraction for 3-d object recognition of industrial parts. In *IEEE-CS Int. Conf. Robotics and Automation*, pages 380–386, New York, NY, 1987. IEEE Press.
- [17] R.M. Haralick. *Surfaces in Range image Understanding*. Springer-Verlag, New York, 1988.
- [18] S.L. Horowitz and T. Pavlidis. Picture segmentation by a direct split-and-merge procedure. In *Proc. 2nd Int. Joint Conf. Pattern Recognition*, pages 424–433, New York, NY, 1974. IEEE Press.
- [19] R. Hummel. Image enhancement by histogram transformations. *Comput. Graphics and Image Processing*, 6:184–195, 1977.
- [20] F. Inanc and J.N. Gray. A CAD interfaced simulation tool for X-ray NDE studies. In D.O. Thompson, editor, *Review of Progress in Quantitative Non-Destructive Evaluation*, volume 9a, pages 391–398. Plenum Press, New York, 1990.
- [21] Precision Visuals Inc. PV-WAVE technical reference manual. Technical report, Precision Visuals Inc., Boulder, CO, 1991.
- [22] A. Khellaf, A. Beghdadi, and H. Dupoisot. Entropic contrast enhancement. *IEEE Trans. Medical Imaging*, 10(4):589–592, December 1991.

- [23] H. Koshimizu, K. Murakami, and K. Hasegawa. On the knowledge based interactive pattern processing system for the recognition of weld defects in radiographic inspection. In *IEEE-CS Int. Conf. Pattern Recognition*, pages 14–17, New York, NY, 1986. IEEE Press.
- [24] D.T. Kuan, A.A. Sawchuk, T.C. Strand, and P. Chavel. Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Trans. PAMI*, PAMI-7(2):165–177, March 1985.
- [25] D.C. Lai, J. Potenza, and K. Verfaillie. Evaluation of image restoration filters for machine classification. *Optical Engineering*, 23(6):794–800, 1984.
- [26] J.S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. PAMI*, PAMI-2(2):165–168, 1980.
- [27] J.S. Lee. Refined noise filtering using local statistics. *Comput. Graphics Image Processing*, 15:380–389, 1981.
- [28] S.U. Lee, S.Y. Chung, and R.H. Park. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, 52:171–190, 1990.
- [29] K.V. Mardia and T.J. Hainsworth. A spatial thresholding method for image segmentation. *IEEE Trans. PAMI*, 10(6):919–927, November 1988.
- [30] M. Oshima and Y. Shirai. A scene description method using three-dimensional information. *Pattern Recognition*, 11:9–17, 1978.
- [31] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man and Cybern.*, SMC-9(1):62–66, January 1979.
- [32] T. Pavlidis and Y.T. Liow. Integrating region growing and edge detection. *IEEE Trans. PAMI*, 12(3):225–233, March 1990.
- [33] A. Pérez and R. Gonzalez. An iterative thresholding algorithm for image segmentation. *IEEE Trans. PAMI*, PAMI-9(6):742–751, November 1987.
- [34] W. Pratt. *Digital Image Processing*. Wiley and Sons, New York, 1978.
- [35] P. Saint-Marc, J.S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Trans. PAMI*, 13(6):514–529, June 1991.

- [36] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, New York, 1988.
- [37] A. Shio. An automatic thresholding algorithm based on an illumination-independent contrast measure. In *Proc. IEEE-CS Conf. Comput. Vision Pattern Recognition*, pages 632–637, New York, NY, 1989. IEEE Press.
- [38] A. Shiozaki. Edge extraction using entropy operator. *Computer Vision, Graphics, and Image Processing*, 36:1–9, 1986.
- [39] J.F. Silverman and D.B. Cooper. Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Trans. PAMI*, 10(4):482–494, July 1988.
- [40] E.M. Siwek and J.N. Gray. Application of the X-ray measurement model to image processing of X-ray radiographs. In D.O. Thompson, editor, *Review of Progress in Quantitative Non-Destructive Evaluation*, volume 12a. Plenum Press, New York, 1993.
- [41] M.M. Skolnick. Application of morphological transformations to the analysis of two-dimensional electrophoretic gels of biological materials. *Computer Vision, Graphics, and Image Processing*, 35:306–332, 1986.
- [42] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. PAMI*, 13(11):1115–1138, November 1991.
- [43] Friedrich M. Wahl. *Digital Image Signal Processing*. Artech House, Norwood, MA, 1987.
- [44] D.C.C. Wang, A.H. Vagnucci, and C.C. Li. Gradient inverse weighted smoothing scheme and the evaluation of its performance. *Comput. Graphics Image Processing*, 15:167–181, 1981.
- [45] R. Wilson and M. Spann. *Image Segmentation and Uncertainty*. Wiley and Sons, New York, 1988.
- [46] J.W. Woods and C.H. Radewan. Kalman filtering in two dimensions. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-29:188–196, April 1981.

- [47] Y. Zheng and J.P. Basart. NDE X-ray image modeling and adaptive filtering considering correlated noise. In D.O. Thompson, editor, *Review of Progress in Quantitative Non-Destructive Evaluation*, volume 7a, pages 813–820. Plenum Press, New York, 1988.

APPENDIX A: NOISE FILTERING PROGRAM

This appendix lists the computer code for the modified adaptive smoothing filter used to suppress noise and preserve flaw signals within a radiographic image. The algorithm suppresses the noise via an adaptive scaling approach. The filtered pixel intensity is the estimate of the local mean added to a scaled version of the original pixel intensity with the local mean subtracted out. The scaling factor is the estimate of the signal variance divided by the sum of the estimated noise and signal variance.

```

FUNCTION masf, in_img, sz

if not keyword_set(in_img) then begin
print, 'FUNCTION masf, in_img, sz'
print, '-----'
print, 'This function will noise filter the input image.'
print, 'This filter is a variation of the adaptive smoothing filter.'
print, 'sz is the dimensions of the square window used for'
print, 'calculating the local statistics.'
print, 'Default:  sz = 3'
return, -1
endif

if not keyword_set(sz) then sz = 3

chksz = size(in_img)
x_dim = chksz(1)
y_dim = chksz(2)

; Create array for result
rslt = double(in_img)

; Make input image dimensions an integer mutiple of sz
x_rgns = x_dim/sz
y_rgns = y_dim/sz
if x_rgns ne x_dim/float(sz) then x_dim = (x_rgns)*sz
if y_rgns ne y_dim/float(sz) then y_dim = (y_rgns)*sz
img = in_img(0:x_dim-1, 0:y_dim-1)

; Find the approximate image of signal only
img = float(img)
avg_org = avg(img)
img = img - avg_org

; This is the approx signal used to calculate signal variance
imgf = adp_smf(img, sobel(img), 6, 5)

; Make the signal zero mean
imgf = imgf - avg(imgf)

; Initialize variables
vrsg = fltarr(x_rgns,y_rgns)
vrns = fltarr(x_rgns,y_rgns)

; Calculate the local statistics
; signal mean - rgno(i,j)
; signal variance - vrsg(i,j)
; noise variance - vrns(i,j)
for i=0,x_rgns-1 do $
  for j=0,y_rgns-1 do begin
    rgns = imgf(i*sz:i*sz+sz-1,j*sz:j*sz+sz-1)
    rgno = img(i*sz:i*sz+sz-1,j*sz:j*sz+sz-1)
    vrsg(i,j) = sigma(rgns)^2
    vrns(i,j) = sigma(rgns-rgno)^2
  endfor

; Smooth the local noise variance
vrns = smootha(median(vrns, 3), 3)

; Correct for the shifting experienced in the rebin function
sz2 = sz/2
vrns_img = shift(rebin(vrns, x_dim, y_dim), sz2, sz2)
vrsg_img = shift(rebin(vrsg, x_dim, y_dim), sz2, sz2)+1

; Subtract out the signal mean
img_mean = adp_smf(imgf, sobel(imgf),15,10)
rgnf_img = img - img_mean

```

```

; Multiply the image signal by local statistics
rgnf_img = rgnf_img*vrsg_img/(vrsg_img+vrns_img)

; Add the signal mean back in
rgnf_img = rgnf_img + img_mean

; Rescale the output to have same range as input
;   and add image offset
rgnf_img = rgnf_img*range(in_img)/range(rgnf_img) + avg_org

; Place filtered image in result
rslt(0,0) = rgnf_img

; Return result
return, rslt
end

```

APPENDIX B: MAIN DRIVER PROGRAM FOR VOSS TECHNIQUE

This computer program is the main driver program for the Variable-Order Surface Segmentation technique described in Section 6.4. The program calls the various procedures that identify the initial seed regions, grow these regions, and approximated the intensities within the grown regions. This program also performs various house keeping task that organize the information concerning the location of the grown regions and the parameters associated with the approximated surface function within these regions.

```

FUNCTION strt_itr, in_img, timg, sd_rgns, sd_lmt, gu, gv
    if not keyword_set(in_img) then begin
    print, ' FUNCTION strt_itr, in_img, timg, sd_rgns, sd_lmt, gu, gv'
    print, ' -----'
    print, ' This function will iteratively grow seed regions found in the'
    print, ' surface type image (timg) until all seed regions gt sd_lmt'
    print, ' have been grown. After each seed region terminates growth,'
    print, ' the region's parameters are recorded in the surface-fit-list.
        return, 1
    endif

; *** INPUT
; in_img - original image
; sd_lmt - minimum number of pixels in a seed region

;*** RETURNED
; rcn_img - Residual image found by reducing effects
;           of geometrical structures

nimg = in_img
mimg = median(nimg, 5)
imsz = size(nimg)
xdim = imsz(1)
ydim = imsz(2)
vnm = [[0,1,0],[1,1,1],[0,1,0]]
flt = [[0,0,1,0,0],[0,1,1,1,0],[1,1,1,1,1],[0,1,1,1,0],[0,0,1,0,0]]
new_rgn = bytarr(xdim, ydim) + 1
grw_cnt = intarr(5)

; Lower noise limit ( no need for higher order surface fit)
lwr_ns_lmt = 10

; Upper noise limit ( don't add pixels with noise larger than this)
ns_lmt = 10

; Upper polynomial order limit for surface fitting
ordr_lmt = 3

; Maximum number of coefficient for describing a surface
mx_cf = 3 +(ordr_lmt eq 2)*3 +(ordr_lmt eq 3)*7 +(ordr_lmt eq 4)*12

; Identify surface types in original img
timg = surf_type(mimg, 2, gu, gv)

; Identify seed regions
sd_rgns = seed_rgns(mimg, timg, sd_lmt)

; Initialize variables for surface regions
sfl_coef = fltarr(mx_cf, 2*max(sd_rgns)+1)
sfl_ordr = bytarr(2*max(sd_rgns)+1)
sfl_rgns = bytarr(xdim, ydim)
sfl_errs = fltarr(xdim, ydim) + 999.0
rcn_img = intarr(xdim, ydim)

; Initialize variable for the current region iteration
rgn_coef = fltarr(mx_cf, 4)
rgn_errs = fltarr(4)

; Number of the current region
rgn_num = 1

; Extract first seed region
c_rgn = next_rgn(sd_rgns, sfl_rgns, timg, rgn_num, sd_lmt)

; Start the iterative variable-order surface segmentation procedure
; - continue until no valid seeds are found

```

```

while total(c_rgn) do begin
    tvscl, c_rgn

; ** Set the surface fit order
    s_ordr = 0
    grw_cnt(*) = 0
    grw_cnt(0) = 2
    rgn_errs(*) = 0
    rgn_errs(0) = 999.0

; ** Find initial surface sit order for seed region
    rgn_parm = {0.0, 31, 0, topo_dim(c_rgn, ar, per), 0, 0, per}
    while (rgn_parm(1) gt lwr_ns_lmt) and (s_ordr lt ord_r_lmt) do begin
        s_ordr = s_ordr + 1
        o_mtrx = crt_mtrx(mimg, c_rgn, s_ordr, o_ysol)
        b_coef = lssfit(o_mtrx, o_ysol, s_ordr, errs)
        appx_rgn = rgn_appx(c_rgn, s_ordr, b_coef)
        rgn_parm(1) = 3*sigma((appx_rgn-nimg)(where(c_rgn)))
        print, rgn_parm(1)
        grw_cnt(s_ordr) = 2
    endwhile
    if rgn_parm(1) gt ns_lmt and rgn_parm(1) lt 1.5*ns_lmt then $
        rgn_parm(1) = ns_lmt -.01

; ** Prepare screen for visual monitoring of regions
    wset,2
    erase
    tv, bytarr(512)+255,0,400
    tv, bytarr(512)+255,0,300
    tv, bytarr(512)+255,0,150
    ngrwth = {0,s_ordr,0}
    wset,0

; ** Grow the current region until highest order is reached or
;    - no more pixels are added

    while (rgn_parm(1) lt ns_lmt) and (s_ordr lt ord_r_lmt+1) $
        and (grw_cnt(s_ordr-1) gt 1) do begin
            a_coef = b_coef
            ngrwth = {0,s_ordr,ngrwth(2)}
            wset, 2
            tv, bytarr(1,512)+255,ngrwth(2),0
            wset, 0

; **** Find piecewise-smooth surface fuction for current region
            repeat begin
                prv_rgn = c_rgn
                if rgn_parm(1) lt ns_lmt then $
                    new_rgn = grower(in_img, c_rgn, a_coef,$
                                    rgn_parm, sfl_errs, ngrwth, gu, gv)
                rgn_grw = total(new_rgn - prv_rgn)
                if (rgn_grw ne 0 and (rgn_parm(1) lt lwr_ns_lmt/2)) or $
                    (rgn_grw gt 10) then begin

; ***** Recalculate coefficients for best fit
                    e_mtrx = crt_mtrx(mimg, new_rgn-c_rgn, s_ordr, e_ysol)
                    o_mtrx = [[o_mtrx],[e_mtrx]]
                    o_ysol = {o_ysol,e_ysol}
                    a_coef = lssfit(o_mtrx, o_ysol, s_ordr, errs)
                endif
                c_rgn = new_rgn
            endrep until not total(new_rgn - prv_rgn)

            rgn_coef(0,s_ordr) = a_coef
            rgn_errs(s_ordr) = rgn_parm(1)
            grw_cnt(s_ordr) = ngrwth(0)

```

```

; **** Check to see if significant growth occurred
; - if it did then increase the order and continue
s_ordr = s_ordr + 1
if (s_ordr lt (ordr_lmt+1) and $
    grw_cnt(s_ordr-1) gt 1) then begin
    o_mtrx = crt_mtrx(mimg, c_rgn, s_ordr, o_ysol)
    b_coef = lssfit(o_mtrx, o_ysol, s_ordr, errs)
    appx_rgn = rgn_appx(c_rgn, s_ordr, b_coef)
    rgn_parm(0) = 0
    rgn_parm(1) = 3*sigma((appx_rgn-nimg)(where(c_rgn)))
    rgn_errs(s_ordr) = rgn_parm(1)
    if rgn_parm(1) gt ns_lmt then rgn_parm(1) = ns_lmt -.01
    print, rgn_parm
    print, 'Fitting higher order surface'
endif
endwhile

; ** Region terminated growth
; ** Lower the order of the fit if less than 2 pixels were added
i = s_ordr
while grw_cnt(i) lt 2 do i = i - 1

; ** Lower the order of the fit if less than 4% growth took place.
if grw_cnt(i) lt .04*total(c_rgn) then i = i - 1

; ** Determine if higher order fit is significantly better
if i lt ord_r_lmt then $
    if rgn_errs(i+1) lt rgn_errs(i)/3. then i=i+1

; ** Update the region's parameters in the surface_fit_list
sfl_ordr(rgn_num) = i
sfl_coef(*,rgn_num) = rgn_coef(*,i)
if i ne s_ordr then appx_rgn = rgn_appx(c_rgn, i, rgn_coef(*,i))
rsdl_err = abs(appx_rgn - nimg)
bst_fit = where(rsdl_err lt sfl_errs and new_rgn gt 0)
rcn_img = rcn_img*(1-c_rgn) + appx_rgn*c_rgn
tvsc1, rcn_img, 1
tvsc1, (nimg-rcn_img)*(rcn_img ne 0), 2
sfl_rgns(bst_fit) = rgn_num
tvsc1, sfl_rgns, 3
sfl_errs(bst_fit) = rsdl_err(bst_fit)

; ** Print info concerning region
print, 'seed region terminated with order ', i
print, '----coefs = ', sfl_coef(*,rgn_num)

; ** Increment number of regions grown
rgn_num = rgn_num + 1

; ** Extract next seed region
c_rgn = next_rgn(sd_rgns, sfl_rgns, timg, rgn_num, sd_lmt)
new_rgn = c_rgn
endwhile

; Save the surface segmentation info for later use
save, filename = 'rcn.out', sfl_ordr, rgn_num, xdim, ydim, sfl_coef, sfl_rgns

; Merge grown regions
mrg_img = fill_holes(nimg, sfl_coef, sfl_rgns eq 0)

rcn_img = rcn_img + mrg_img

; Subtract reconstructed image from original and return result
return, float(nimg) - rcn_img
end

```


APPENDIX C: SURFACE TYPE CLASSIFICATION PROGRAM

This appendix list the code for identify the topological surface types associated with each pixel in the original image. The resulting image contains eight different intensity levels that correspond to one of the eight possible topological surface types as described in Section 6.4.

```

FUNCTION surf_type, in_img, num_levs, gu, gv
    if not keyword_set(in_img) then begin
print, ' FUNCTION prim_seg, in_img, num_levs, gu, gv'
print, ' -----'
print, ' This function will attempt to classify each pixel into one'
print, ' of 8 different surface type labels found using the slope and'
print, ' and curvature. Num_levs is the # of times to subsample in_img'
return, 1
endif

; *** INPUT
; in_img - Original image
; num_levs - Number of times to subsample original image

; *** OUTPUT
; gu - Horizontal gradient image of original
; gv - Vertical gradient image of original

; *** RETURNED
; timg - Surface type labeled image

nimg = in_img
img = float(in_img)
imsz = size(nimg)
xdim = imsz(1)
ydim = imsz(2)
if not keyword_set(num_levs) then num_levs = 2
qlcvs = num_levs-1.
lcd = fix(2^qlcvs)
if xdim/lcd ne float(xdim)/lcd then nxdim = (xdim/lcd + 1)*lcd
if ydim/lcd ne float(ydim)/lcd then nydim = (ydim/lcd + 1)*lcd
if keyword_set(nxdim) or keyword_set(nydim) then $
    img = congrid(nimg, nxdim, nydim)

vnm = [[0,1,0],[1,1,1],[0,1,0]]
sqr = [[1,1,1],[1,1,1],[1,1,1]]

; Determine the dimensions of the subsampled image
ss_xdim = xdim/2^(num_levs-1)
ss_ydim = ydim/2^(num_levs-1)

; Reduce the image to conserve computation time
; and eliminate noise
st_img = rebin(img,ss_xdim,ss_ydim)

; Determine the size of the smoothing template
tmp_sz = min([ss_xdim, ss_ydim])/15 -1

; Calculate the smoothed image
sbl = sobel(img)
simgl = adp_smf(img, sbl, 6, 5)

; Create least-squares derivative estimation window operators
d0 = 1./7*[1,1,1,1,1,1,1]
d1 = 1./28*[-3, -2, -1, 0, 1, 2, 3]
d2 = 1./84*[5, 0, -3, -4, -3, 0, 5]

du = d1#d0
dv = d0#d1
duu = d2#d0
dvv = d0#d2
duv = d1#d1

; Calculate the partial derivative estimate images
gu = convol(simgl, du)
gv = convol(simgl, dv)

```

```

guu = convol(simg1, duu)
gvv = convol(simg1, dvv)
guv = convol(simg1, duv)

; Compute the mean (H) and Gaussian (K) curvature images

th = 0.03
tk = 0.015
h = ((1+gv^2)*guu+(1+gu^2)*gvv - 2*gu*gv*guv)/( 2*sqrt(1+gu^2+gv^2)^3)
hs = smootha(h, tmp_sz)
k = guu*gvv-guv^2/(1+gu^2+gv^2)^2
ks = smootha(k, tmp_sz)

; Compute the surface type image based on the sign of H and K
timg = 1 + 3*(1 + (abs(hs) gt th)*(-1)^(hs lt 0)) + $
      (1 - (abs(ks) gt tk)*(-1)^(ks lt 0))

; Return the surface type classified image
return, timg
end

```

APPENDIX D: INITIAL SEED REGION IDENTIFICATION PROGRAMS

The programs used to obtain an initial guess of the seed regions representing the relevant geometrical structures within the image are listed here. The main seed identification program is *seed_rgns()*. This program calls upon the others to perform tasks that are required to accurately identify these seed regions. These programs are mentioned in Section 6.5.1.

```

FUNCTION seed_rgns, in_img, timg, sd_lmt, gm_lmt, parm

    if not keyword_set(in_img) then begin
print, ' FUNCTION seed_rgns, in_img, timg, sd_lmt, gm_lmt'
print, ' -----'
print, ' This function will identify seed regions within the image by'
print, ' processing and grouping the surface type image (timg).'
print, ' The function will return an image of seed regions with more'
print, ' than sd_lmt pixels.'
        return, 1
    endif

; *** INPUT
; in_img - Original image
; timg - Image containing surface type classification results
; sd_lmt - Minimum number of pixels needed to be a seed region
; gm_lmt - Minimum topological descriptor value required for seeds

; *** RETURNED
; fin_rslt - Image containing valid seed regions

nimg = in_img
imsz = size(nimg)
xdim = imsz(1)
ydim = imsz(2)
vnm = [[0,1,0],[1,1,1],[0,1,0]]
flt = shift(dist(3), 1,1) lt 1.1

; Create an image for preliminary results
rslt = intarr(xdim,ydim)

; Create an image for final results
fin_rslt = intarr(xdim,ydim)
rgn_sz = intarr(pxdim,pydim)

; Determine number of pixels represented in one pixel from timg
rgn_scl = 4^(xdim/pxdim - 1)

; Identify seed regions in surface type image by removing
; small interior regions
rgns = bytarr(pxdim, pydim)
for i=1,9 do rgns= rgns + erode(median(median(timg eq i,5),5)*i,flt)

; Fix problem around the edges of image
rgns(*,1) = 0
rgns(*,pydim-2) = 0
rgns(pxdim-2,*) = 0
rgns(1,*) = 0

; Identify each candidate region with a unique number
nrgns = f_isl4n(rgns gt 0, sd_lmt/rgn_scl+1)

; Number the seed regions according to size
pix_rgn = histogram(nrgns)
num_rgns = max(nrgns)
pix_rgn = pix_rgn(1:num_rgns)
sort_sz = sort(pix_rgn) + 1
for i=0, num_rgns-1 do $
    rgn_sz = rgn_sz + (nrgns eq sort_sz(i))*(num_rgns-i)

; Iterative method for identifying reliable seed regions
rgn_num = 0
for i=1, num_rgns do begin
    rgn = rslt eq i
    area = sd_lmt + 1
    repeat begin
        s_rgn = rgn

```

```

d1 = gm_lmt-1
area = sd_lmt + 1
print, 'region: ', i
  while d1 lt gm_lmt and area gt sd_lmt do begin
    s_rgn = erode(s_rgn, vnm)
    lrgn = largest(s_rgn, mx_sz)
    tvscl, lrgn, pxdim, pydim
    lrgn = xpnd(lrgn, rgn)
    print, 'max :', mx_sz
    tvscl, lrgn
    d1 = topo_dim(lrgn, area)
    print, d1
  endwhile
print, 'area, sd_lmt', area, sd_lmt
if area gt sd_lmt then begin
  rgn_num = rgn_num + 1
  sdrgn = shrink(lrgn, sd_lmt, gm_lmt)
  x_rgn = xpnd(sdrgn, rgn)
  tvscl, sdrgn+lrgn+x_rgn, 0, pydim
  fin_rslt = fin_rslt + x_rgn*rgn_num
  tvscl, fin_rslt gt 0, 0
endif
rgn = (rgn - lrgn) eq 1
tvscl, rgn, 128, 0
endrep until area lt sd_lmt
endfor

; Display final and preliminary results
tvscl, [fin_rslt gt 0, rslt gt 0]

; Return image of the valid seed regions
return, fin_rslt
end

```

```

FUNCTION xpnd, rgn, bndry

    if not keyword_set(rgn) then begin
print, ' FUNCTION xpnd, rgn, bndry'
print, ' -----'
print, ' This function will grow rgn out to bndry under the condition'
print, ' that its topological descriptor does not decrease.'
        return, 1
    endif

; *** INPUT
; rgn - Binary image showing the smaller region to be expanded
; bndry - Binary image showing

; *** RETURN
; n_rgn - Binary image showing the expanded region

vnm = [[0,1,0],[1,1,1],[0,1,0]]
n_rgn = rgn

    area = total(n_rgn)
    d_rgn = dilate(n_rgn, vnm)
    perim = total(d_rgn - n_rgn)
    new_td = area/(perim*perim)
    old_td = new_td - 1

    while new_td gt old_td do begin
        n_rgn = d_rgn and bndry
        area = total(n_rgn)
        d_rgn = dilate(n_rgn, vnm)
        perim = total(d_rgn - n_rgn)
        old_td = new_td
        new_td = area/(perim*perim)
    endwhile

; Return expanded region

return, n_rgn
end

```

```

FUNCTION largest, rgns, sz

    if not keyword_set(rgns) then begin
print, ' FUNCTION largest, rgns, sz'
print, ' -----'
print, ' This function will return the largest adjoined region within'
print, ' the input image rgns.  sz is the number of pixels in it.'
        return, 1
    endif

; *** INPUT
; rgns - binary image showing the pixels to consider

; *** OUTPUT
; sz - area of the largest region in rgns

vnm = [[0,1,0],[1,1,1],[0,1,0]]

; if no pixel are on in rgns then return rgns
; if not total(rgns) then return, rgns

; label each 4-connected region with a unique number
i_rgn = f_isl4n(rgns, 1)

; count the number of pixels in each unique region
pix_rgn = histogram(i_rgn)
n_rgns = max(i_rgn)

; get the largest area
sz = max(pix_rgn(1:n_rgns))
lrgst = !c + 1

; return the largest unique region
return, i_rgn eq lrgst
end

```



```

FUNCTION f_isl4n, img, lmt_sz

if not keyword_set(img) then begin
print, ' FUNCTION f_isl4n, img, lmt_sz'
print, ' -----'
print, ' This function will label the individual islands in the binary
print, ' image (img) that contain at least (lmt_sz) pixels. The'
print, ' function returns an image with the islands labeled with '
print, ' consecutive numbers.'
print, ' 4-nearest neighbors connectivity is used.'
return, 1
endif

; *** INPUT
; img      - binary image showing the regions to be identified
; lmt_sz   - Minumum number of pixels required to make an island

; *** RETURNED
; rslt     - image of consecutively numbered islands

; Convert input image (img) into binary image
if n_elements(where(histogram(img))) gt 2 then begin
    print, ''
    print, 'The input image must be binary!!'
return, -1
endif

;define variables
;store isl_elements here
xbndry = intarr(90000)
ybndry = intarr(90000)
dims = size(img)
ox_dim = dims(1)
oy_dim = dims(2)
n_isl = 0
minval = min(img)

; Expand image by one pixel for border
e_img = bytarr(ox_dim+2, oy_dim+2)
e_img(1,1) = img
fi_img = e_img
fi_img(*,*) = 0
x_dim = ox_dim+2
y_dim = oy_dim+2

b_img = (e_img lt 100) and e_img
nzvals = where(b_img)

; Start the find islands procedure
repeat begin

    starty = nzvals(0)/x_dim
    startx = nzvals(0)-x_dim*starty
    x = startx
    y = starty

; The search matrix looks like this
; x is the current location
;
;      1
;    4 x 2
;      3

; The first direction to check after moving is determined
; by the chart below. This causes the program to trace out
; the regions in a clockwise direction.
; got there by ----- then start with

```

```

; 1 4
; 2 1
; 3 2
; 4 3

; Initialize variables for each new island
l = 1
xbndry(0:1) = x
ybndry(0:1) = y
n_isl = n_isl + 1
fi_img(x,y) = n_isl
ddend = 0

repeat begin

if b_img(x,y-1) then y=y-1 else $
if b_img(x+1,y) then x=x+1 else $
if b_img(x,y+1) then y=y+1 else $
if b_img(x-1,y) then x=x-1 else $
begin ; dead end - back up
ddend = 1
l = l - 1
x = xbndry(l)
y = ybndry(l)
endelse

if not ddend then l = l + 1
ddend = 0
xbndry(l) = x
ybndry(l) = y
fi_img(x,y) = n_isl
b_img(x,y) = 0
endrep until l eq 0

new_isl= where(fi_img eq n_isl)

; Check to see if island contains minimum number of pixels
if n_elements(new_isl) lt lmt_sz then begin
n_isl = n_isl -1
fi_img(new_isl) = 0
endif
nzvals = where(b_img)
endrep until n_elements(nzvals) eq 1

; Return image of numbered islands
return, fi_img(1:ox_dim, 1:oy_dim)

end

```

APPENDIX E: REGION GROWING PROGRAMS

The appendix lists the programs used to grow the seed regions into piecewise-smooth surface function that approximate the intensity variations due to geometrical structures. The main program for growing the region is *grower()*. This program will iteratively grow a seed based on the constraints discussed in Section 6.5.2 and implemented in *rgn_grwr()*, *topo_dim()* and *nrml()*. When a region has terminates growth, *grower()* returns the grown region and the corresponding parameters that describe the approximating surface functions to *strt_itr()*.

```

FUNCTION grower, in_img, c_rgn, coef, rgn_parm, err_img, ngrwth, gu,gv

    if not keyword_set(in_img) then begin

print,'FUNCTION grower, in_img, c_rgn, coef, rgn_parm, $
                                err_img, ngrwth, gu, gv'
print,'-----'
print,' This function will iteratively grow c_rgn using coef'
print,' from the surface fit of new_rgn and the std dev found from'
print,' this fit. After the region terminates growth, the grown'
print,' region is returned.'
        return, 1
    endif

; *** INPUT
; in_img - original image
; c_rgn - The region being grown
; s_ordr - The order of the polynomial : max value of c_rgn
; coef - The last coefficients calculated for that region
; rgn_parm - Array of previous and current region parameters
;          (0) = Previous fitting error.
;          (1) = Current fitting error.
;          (2) = Previous topological descriptor.
;          (3) = Current topological descriptor.
;          (4) = Number of pixels added last time.
;          (5) = Number of pixels added time before that.
;          (6) = Perimeter of current region.
; err_img - Image containing errors from best fits to actual surface
; ngrwth - Array
;          (0) = Running total of pixels added to seed
;          (1) = s_ordr - The order of the polynomial
;          (2) = total number of growth iterations*2
; gu - Horizontal gradient of original image
; gv - Verticle gradient of original image

nimg = in_img
imsz = size(nimg)
xdim = imsz(1)
ydim = imsz(2)
mrk = [[1,1],[1,1]]

; Upper noise limit ( don't add pixels with noise larger than this)
ns_lmt = 10
s_ordr = ngrwth(1)

; * Approximate the current region to determine fitting error
appx_rgn = rgn_appx(c_rgn, s_ordr, coef)
tvsc1, appx_rgn, 0, ydim

; * Get previous fitting error
rgn_parm(0) = rgn_parm(1)

; * Get current fitting error
rgn_parm(1) = 3*sigma((appx_rgn-nimg)(where(c_rgn)))
if rgn_parm(4) lt 40 then rgn_parm(1) = 1.08*rgn_parm(0)
if rgn_parm(1) lt 1.5 then rgn_parm(1) = 2.0
if rgn_parm(1) gt ns_lmt then rgn_parm(1) = ns_lmt-.01
print, 'Limit for adding to rgn :', rgn_parm(1)
new_rgn = c_rgn

; * Allow tst_rgn to be everywhere that new_rgn is not.
tst_rgn = byte(1-new_rgn)

; * Set a limit on the maximum number of new growths allowed
grw_lmt = total(c_rgn)/5.0
new_pix = 0

```

```

; * Grow region until no new growths occur or the number of
; * new growths exceeds the growth limit grw_lmt.
repeat begin
    grwths = 0
    g_rgn = new_rgn
    new_rgn = rgn_grwr(nimg, g_rgn, s_ordr, coef, rgn_parm(1), $
        tst_rgn, err_img, grwths, gu, gv)
    new_pix = new_pix + grwths
endrep until not grwths or (new_pix gt grw_lmt)
rgn_parm(4) = new_pix

; * Fill in the small noisy holes that occurred in the growing process
new_rgn = fill(new_rgn)
tvscl, new_rgn

; * Update the topological descriptors for the new region
rgn_parm(2) = rgn_parm(3)
rgn_parm(3) = topo_dim(new_rgn, area, per)

; * Measure change in topological descriptor (ctd)
ctd = new_pix/(per-rgn_parm(6))^2

; * Rate of growth
rg = new_pix/area
ans = 0

; * Rate of decay of the topological descriptor
rtd = (rgn_parm(2)-rgn_parm(3))/rgn_parm(2)
grw_tst = (new_pix + rgn_parm(4) + rgn_parm(5)) gt 10

; * Plot chart to monitor progress
plot_chart, ngrwth(2), rg, rtd, rgn_parm(3), rgn_parm(1), mrk

; * Monitor the regions size and shape to detect undesirable growth
if not grw_tst then ans = 1 $
else if rgn_parm(1) lt ns_lmt/2 then ans = 3 $
else if rgn_parm(3) lt 50 then begin
    print, 'Rate :',rg, rtd
    if rg lt rtd then begin
        tvscl, 2*new_rgn - c_rgn
        if ctd lt .2 then $
            if (rgn_parm(2)-rgn_parm(3)) gt 1.5 and not grw_tst $
                then ans = 2 $
            else if 1.5*rg lt rtd then ans = 1 $
            else if not grw_tst then ans = 1
        endif
        print, 'ANS :',ans
    endif
endif
if ans eq 3 then rgn_parm(1) = 1.1*rgn_parm(1) $
else if ans gt 0 then begin
    rgn_parm(3) = rgn_parm(2)
    per = rgn_parm(6)
    new_rgn = c_rgn
    new_pix = 0
endif

; * Update the current region's parameters
rgn_parm(6) = per
rgn_parm(5) = rgn_parm(4)
rgn_parm(4) = new_pix
ngrwth(0) = ngrwth(0) + new_pix
ngrwth(2) = ngrwth(2) + 4

; * Return the region
return, new_rgn
end

```

```

FUNCTION rgn_grwr, in_img, o_rgn, podr, coef, err_lmt, $
    tst_rgn, err_img, num_added, gu, gv

    if not keyword_set(in_img) then begin
        FUNCTION rgn_grwr, in_img, o_rgn, podr, coef, err_lmt, $
        print, '          tst_rgn, err_img, num_added'
        print, '-----'
        print, ' This function will determine which pixels neighboring o_rgn'
        print, ' satisfy an error test for a bivariate polynomial of order'
        print, ' (podr).'
        return, 1
    endif

; *** INPUT
; in_img - original image
; o_rgn - region in which to grow
; podr - order of surface fit (1,2 or 3)
; coef - coefficients found from previous surface fit
; gu - Horizontal gradient of original image
; gv - Vertical gradient of original image

; *** INPUT AND OUTPUT
; err_lmt - allowable error between approximated and actual surface
; tst_rgn - region where new pixels have not been tested
; err_img - Image containing error for best fit to actual surface

; *** OUTPUT
; num_added - Total number of points added to region this time
; *** RETURNED
; n_rgn - binary image showing the grown region

nimg = in_img
imsz = size(nimg)
xdim = imsz(1)
ydim = imsz(2)
vnm = [[0,1,0],[1,1,1],[0,1,0]]

e_rgn = bytarr(xdim, ydim)
g_rgn = bytarr(xdim, ydim)

; Identify neighboring pixels that haven't been tested yet (n_pix)
o_rgn = o_rgn gt 0
do_rgn = dilate(o_rgn, vnm)
n_pix = do_rgn and tst_rgn
tn_pix = total(n_pix)
if tn_pix lt 10 then print, '## ', tn_pix
if tn_pix lt 7 then return, o_rgn
tst_rgn = tst_rgn and (not do_rgn)

; Get the element numbers for the pixels in n_pix
elnums = where(n_pix)
if elnums(0) eq -1 then begin
    num_added = 0
    return, o_rgn
endif

; How many pixels in n_pix
szrg = n_elements(elnums)
; Determine the corresponding image coordinates
x_c = lonarr(szrg)
y_c = lonarr(szrg)
for j=0, szrg-1 do begin
    y_c(j) = elnums(j)/xdim
    x_c(j) = elnums(j) mod xdim
endfor

; Create the regression matrix

```

```

; Which order polynomial
mdim = 2 + (podr eq 2)*3 + (podr eq 3)*7 + (podr eq 4)*12
mtrx = dblarr(mdim+1, szrg)
mtrx(0,*) = 1.0

; 1st order fit
for j=0, szrg-1 do begin
    mtrx(1,j) = x_c(j) ; a1*x
    mtrx(2,j) = y_c(j) ; a2*y
endfor

; 2nd order fit
if podr gt 1 then $
    for j=0, szrg-1 do begin
        mtrx(3,j) = x_c(j)*y_c(j) ; a3*xy
        mtrx(4,j) = x_c(j)^2      ; a4*x^2
        mtrx(5,j) = y_c(j)^2      ; a5*y^2
    endfor

; 3rd order fit
if podr gt 2 then $
    for j=0, szrg-1 do begin
        mtrx(6,j) = mtrx(3,j)*y_c(j) ; a6*xy^2
        mtrx(7,j) = mtrx(3,j)*x_c(j) ; a7*yx^2
        mtrx(8,j) = mtrx(4,j)*x_c(j) ; a8*x^3
        mtrx(9,j) = mtrx(5,j)*y_c(j) ; a9*y^3
    endfor

; 4th order fit
if podr gt 3 then $
    for j=0, szrg-1 do begin
        mtrx(10,j) = mtrx(8,j)*y_c(j) ; a10*yx^3
        mtrx(11,j) = mtrx(4,j)*mtrx(5,j) ; a11*x^2y^2
        mtrx(12,j) = mtrx(9,j)*x_c(j) ; a12*xy^3
        mtrx(13,j) = mtrx(8,j)*x_c(j) ; a13*x^4
        mtrx(14,j) = mtrx(9,j)*y_c(j) ; a14*y^4
    endfor

; Evaluate the fit for the neighboring pixels
errs = double(in_img(elnums)) - coef#mtrx

; Compare the fit to previous fitted regions
bsfit = abs(errs) lt err_img(elnums)

; Create binary image for pixels satisfying fit
; Add these to the region no matter the normal constraints
g_rgn(x_c, y_c) = (abs(errs) lt .3*err_lmt) * bsfit

; Check these for normal constraints
e_rgn(x_c, y_c) = (abs(errs) lt err_lmt) * bsfit

; Create binary image for compatible pixels satisfying normal test
en_rgn = nrml(e_rgn-g_rgn, podr, coef, gu, gv)

; Define new region for surface fit
n_rgn = o_rgn + en_rgn + g_rgn

; These pixels were compatible but didn't satisfy normal criterion
tvsc1, e_rgn-en_rgn, 0

; Count number of pixel added this iteration
num_added = total(en_rgn+g_rgn)

; Return the grown region
return, n_rgn
end

```

```

FUNCTION topo_dim, rgn, area, perim

    if not keyword_set(rgn) then begin
print, ' FUNCTION topo_dim, rgn, area, perim'
print, ' -----'
print, ' This function will return the topological dimension of rgn.'
        return, 1
    endif

; *** INPUT
; rgn - binary image specifying region to be measured

; *** OUTPUT
; area - Calculated area of the region
; perim - Calculated perimeter of the region

; *** RETURNED
; td - topological descriptor value for region

vnm = [[0,1,0],[1,1,1],[0,1,0]]

e_rgn = erode(rgn, vnm)
perim = total(rgn - e_rgn)
area = total(rgn)
td = area*1000/(perim*perim)

; Return the calculated topological descriptor
return, td
end

```



```

FUNCTION nrml, img, podr, coefs, gu, gv

if not keyword_set(img) then begin
print, 'FUNCTION nrml, img, podr, coefs, gu, gv'
print, '-----'
print, ' This function checks the compatible pixels in img for'
print, ' surface normal criteria with the approximated surface.'
print, ' The function returns an image containing satisfying pixels.'
return, -1
endif

; *** INPUT
; img - Binary image where surface normal criteria should be checked
; podr - Order of the polynomial used to approximate the surface
; coef - Coefficients used to approximate the surface
; gu - Horizontal gradient of original image
; gv - Vertical gradient of original image

; *** RETURN
; rslt - Binary image where checked pixels meet criterion

imsz = size(img)
xdim = imsz(1)
ydim = imsz(2)

; Create image to store the results
rslt = bytarr(xdim, ydim)
; Threshold angle used to check surface normal agreement
theta = 15 + podr*10
; Identify location of pixels that need to be checked
elnums = where(img)
if elnums(0) eq -1 then return, rslt
; How many pixels need to be checked
szrg = n_elements(elnums)
; Determine the corresponding elements in gu and gv
gcu = gu(elnums)
gcv = gv(elnums)
; Determine the corresponding image coordinates
x_c = lonarr(szrg)
y_c = lonarr(szrg)
for j=0, szrg-1 do begin
    y_c(j) = elnums(j)/xdim
    x_c(j) = elnums(j) mod xdim
endfor

; Calculate the estimated surface normals for the candidate pixels
gpu = replicate(coefs(1), szrg)
gpv = replicate(coefs(2), szrg)
if podr gt 1 then begin
    gpu = gpu + coefs(3)*y_c + 2.*coefs(4)*x_c
    gpv = gpv + coefs(3)*x_c + 2.*coefs(5)*y_c
endif
if podr gt 2 then begin
    gpu = gpu + coefs(6)*y_c^2 + 2*coefs(7)*x_c*y_c + 3*coefs(8)*x_c^2
    gpv = gpv + coefs(7)*x_c^2 + 2*coefs(6)*x_c*y_c + 3*coefs(9)*y_c^2
endif

tst = ((gcu-gpu)^2 + (gcv-gpv)^2 + (gcu*gpv - gcv*gpu)^2) / $
      ((1+gcu^2+gcv^2)*(1+gpu^2+gpv^2))

; Find pixel locations that meet surface normal criterion
rslt(x_c, y_c) = tst lt sin(theta/!radeg)^2

; Return the image showing the pixels that satisfy the normal test
return, rslt
end

```

APPENDIX F: LEAST SQUARES FITTING PROGRAMS

This appendix lists the various programs used to approximate the surface intensities within the original image. The first program, *crt_mtrx()*, lists the code that transforms the image data into the matrix form that is used by *lssfit()* to calculate the least squares fit coefficients. The last program, *rgn_appx()*, uses these coefficients to calculate the approximated surface intensities for a specified region.

```

FUNCTION crt_mtrx, in_img, rgn, podr, y_sol

    if not keyword_set(in_img) then begin
print, ' FUNCTION crt_mtrx, in_img, rgn, podr, y_sol'
print, ' -----'
print, ' This function will extract the coordinates of the pixels lying'
print, ' in (rgn) and return an array representing the data for the'
print, ' polynomial order being fit (podr).'
```

```

        return, 1
    endif

nimg = in_img
imsz = size(nimg)
xdim = imsz(1)
ydim = imsz(2)
zro = long(0)

; Get the element numbers for the pixels in rgn
elnums = where(rgn)
; How many pixels in rgn
szrg = n_elements(elnums)

; Determine the cooresponding coordinates
x_c = intarr(szrg)
y_c = intarr(szrg)
for j=zro, szrg-1 do begin
    y_c(j) = elnums(j)/xdim
    x_c(j) = elnums(j) mod xdim
endfor

; Create the regression matrix
; Which order polynomial
mdim = 2 + (podr eq 2)*3 + (podr eq 3)*7 + (podr eq 4)*12
mtrx = dblarr(mdim+1, szrg)
mtrx(0,*) = 1.0

; 1st order fit
for j=zro, szrg-1 do begin
    mtrx(1,j) = x_c(j) ; a1*x
    mtrx(2,j) = y_c(j) ; a2*y
endfor
; 2nd order fit
if podr gt 1 then $
    for j=zro, szrg-1 do begin
        mtrx(3,j) = x_c(j)*y_c(j) ; a3*xy
        mtrx(4,j) = x_c(j)^2      ; a4*x^2
        mtrx(5,j) = y_c(j)^2      ; a5*y^2
    endfor
; 3rd order fit
if podr gt 2 then $
    for j=zro, szrg-1 do begin
        mtrx(6,j) = mtrx(3,j)*y_c(j) ; a6*xy^2
        mtrx(7,j) = mtrx(3,j)*x_c(j) ; a7*yx^2
        mtrx(8,j) = mtrx(4,j)*x_c(j) ; a8*x^3
        mtrx(9,j) = mtrx(5,j)*y_c(j) ; a9*y^3
    endfor
; 4th order fit
if podr gt 3 then $
    for j=zro, szrg-1 do begin
        mtrx(10,j) = mtrx(8,j)*y_c(j) ; a10*yx^3
        mtrx(11,j) = mtrx(4,j)*mtrx(5,j) ; a11*x^2y^2
        mtrx(12,j) = mtrx(9,j)*x_c(j) ; a12*xy^3
        mtrx(13,j) = mtrx(8,j)*x_c(j) ; a13*x^4
        mtrx(14,j) = mtrx(9,j)*y_c(j) ; a14*y^4
    endfor

```

```
; Create array of solutions for each set of coordinates
y_sol = double(in_img(elnums))

return, mtrx
end
```

```

FUNCTION lssfit, mtrx, yfit, podr, errs

    if not keyword_set(mtrx) then begin
print, ' FUNCTION lssfit, mtrx, yfit, podr, errs'
print, ' -----'
print, ' This function will perform a linear least squares fit to yfit
print, ' using the data in mtrx.Y will be estimated by a 2-dimensional
print, ' polynomial of order(podr). The fit parameters are in coef and
print, ' the error measurement of the fit is in errs.'
        return, 1
    endif

; *** INPUT
; mtrx - Input array containing the matrix values for each (x,y) point
; yfit - Input array of solutions to the polynomial equations
; podr - Order of the polynomial (1,2, or 3)
; errs - Array containing goodness of fit values

; *** RETURNED
; coef - array of least squares fit coefficients

; Use uniform weighting function
wt = dblarr(n_elements(yfit)) + 1

    len = (size(mtrx))(1)-1

; Calculate the LS fit coefficients
coef = regress(mtrx(1:len, *), yfit, wt, yf, a0, sig, ft, r, rm, c)
coef = [a0, reform(coef)]

; Create array of error measurements
errs = [ft, c]

return, coef
end

```

```

FUNCTION rgn_appx, o_rgn, podr, coef
    if not keyword_set(o_rgn) then begin
    print, ' FUNCTION rgn_appx, o_rgn, podr, coef'
    print, ' -----'
    print, ' This function will evaluate the bivariate polynomial for'
    print, ' the specified region (o_rgn) using the coefficients (coef)'
    print, ' and the polynomial order (podr).'
    return, 1
    endif

; *** INPUT
; o_rgn - Binary image of region to approximate
; podr - Polynomial order to use in approximation
; coef - Coefficients to use in approximation

; *** RETURNED
; e_rgn - Image containing approx intensities for specified region

imsz = size(o_rgn)
xdim = imsz(1)
ydim = imsz(2)

e_rgn = fltarr(xdim, ydim)

; Get the element numbers for the pixels in the specified region
elnums = where(o_rgn)
if elnums(0) eq -1 then return, o_rgn

; How many pixels in the specified region
szrg = n_elements(elnums)

; Determine the cooresponding coordinates
x_c = lonarr(szrg)
y_c = lonarr(szrg)
for j=0l, szrg-1 do begin
    y_c(j) = elnums(j)/xdim
    x_c(j) = elnums(j) mod xdim
endfor

; Create the regression matrix
; Which order polynomial
mdim = 2 + (podr eq 2)*3 + (podr eq 3)*7 +(podr eq 4)*12
mtrx = dblarr(mdim+1, szrg)
mtrx(0,*) = 1.0

; 1st order fit
for j=0l, szrg-1 do begin
    mtrx(1,j) = x_c(j) ; a1*x
    mtrx(2,j) = y_c(j) ; a2*y
endfor
; 2nd order fit
if podr gt 1 then $
    for j=0l, szrg-1 do begin
        mtrx(3,j) = x_c(j)*y_c(j) ; a3*xy
        mtrx(4,j) = x_c(j)^2 ; a4*x^2
        mtrx(5,j) = y_c(j)^2 ; a5*y^2
    endfor
; 3rd order fit
if podr gt 2 then $
    for j=0l, szrg-1 do begin
        mtrx(6,j) = mtrx(3,j)*y_c(j) ; a6*xy^2
        mtrx(7,j) = mtrx(3,j)*x_c(j) ; a7*yx^2
        mtrx(8,j) = mtrx(4,j)*x_c(j) ; a8*x^3
        mtrx(9,j) = mtrx(5,j)*y_c(j) ; a9*y^3
    endfor

```

```

; 4th order fit
if podr gt 3 then $
  for j=01, szrg-1 do begin
    mtrx(10,j) = mtrx(8,j)*y_c(j) ; a10*yx^3
    mtrx(11,j) = mtrx(4,j)*mtrx(5,j) ; a11*x^2y^2
    mtrx(12,j) = mtrx(9,j)*x_c(j) ; a12*xy^3
    mtrx(13,j) = mtrx(8,j)*x_c(j) ; a13*x^4
    mtrx(14,j) = mtrx(9,j)*y_c(j) ; a14*y^4
  endfor

; Use the proper number of coefficients
ntrms = 2 + (podr eq 2)*3 + (podr eq 3)*7 +(podr eq 4)*12
coef = coef(0:ntrms)

; Evaluate the function for the specified region
f_int = coef#mtrx

; Create image containing approx intensities for specified region
e_rgn(x_c, y_c) = f_int

return, e_rgn
end

```

APPENDIX G: SEED EXTRACTION PROGRAM

This appendix lists the program that is called by *strt_itr()* to obtain the next seed region to grow in the region growing process. The program finds the largest remaining seed region that does not occupy an area already covered by the previously grown regions.


```

FUNCTION next_rgn, sd_rgns, fit_rgns, rgn_num, sd_lmt

    if not keyword_set(sd_rgns) then begin
print, ' FUNCTION next_rgn, sd_rgns, fit_rgns, rgn_num, sd_lmt'
print, ' -----'
print, ' This function will extract the next seed for region growing'
print, ' from the image of valid seed regions(sd_rgns). The next seed'
print, ' must be larger than sd_lmt and not reside in fit_rgns.'
        return, 1
    endif

; *** INPUT
; sd_rgns - image of valid seed regions
; fit_rgns - Binary image showing pixels covered by previously
;           grown regions
; rgn_num - Number of seed regions already grown
; sd_lmt - Limit for minimum size of seed regions

; *** RETURNED
; nxt_rgn - the next valid seed region to be grown

vnm = [[0,1,0],[1,1,1],[0,1,0]]

; Find remaining seed regions that are not within grown regions
rm_sd = sd_rgns*(sd_rgns gt rgn_num-1)*(fit_rgns eq 0)
if total(rm_sd) eq 0 then return, rm_sd

; Find largest remaining seed gt sd_lmt pixels
pix_rgn = histogram(rm_sd)
n_rgns = max(rm_sd)

; Find the largest number of pixels in any one seed
lrgst = max(pix_rgn(1:n_rgns))
if lrgst lt sd_lmt then return, 0
lrgst1 = !c + 1
nxt_rgn = rm_sd eq lrgst1

; if seed is too large then erode it
while total(nxt_rgn) gt 2*sd_lmt do begin
    r_rgn = erode(nxt_rgn, vnm)
    i_rgn = f_isl4n(r_rgn, 1)
    pix_rgn = histogram(i_rgn)
    n_rgns = max(i_rgn)
    lrgst = max(pix_rgn(1:n_rgns))
    lrgst = !c + 1
    nxt_rgn = largest(r_rgn)
endwhile

; if seed was erode too much then dilate it
while total(nxt_rgn) lt sd_lmt do $
    nxt_rgn = dilate(nxt_rgn, vnm)
    if total(rm_sd eq lrgst1) gt total(nxt_rgn) then rgn_num = rgn_num-1

;Return the next valid seed region
return, nxt_rgn
end

```

APPENDIX H: REGION MERGING PROGRAM

This program was design to fill the gaps remaining between grown regions. The program determines which of the grown regions neighboring the gap would provide the best fit to the original intensities within each gap.

```

FUNCTION fill_holes, f_rgns, in_img, f_coef, hls
    if not keyword_set(f_rgns) then begin
        print, ' FUNCTION fill_holes, f_rgns, in_img, f_coef, hls'
        print, ' -----'
        print, ' This function will fill in the holes with the neighboring'
        print, ' surface that best matches the void region.'
        print, ' The function then returns the completed reconstructed image.'
        return, 1
    endif

; *** INPUT
; f_rgns - region labeled image
; in_img - original image
; f_coef - list of coefficients found for each region
; hls    - binary image showing where gaps exists
; *** RETURNED
; rcn_img - reconstructed image containing approximated intensities
;          for the gaps

nimg = float(in_img)
sz = size(nimg)
xdim = sz(1)
ydim = sz(2)
szlmt = 30
vnm = [[0,1,0],[1,1,1],[0,1,0]]
; Define array to label pixels with a number corresponding
; to the region it resides in
fin_rgns = f_rgns

rcn_img = fltarr(xdim, ydim)
; Find the location of pixels not belong to a grown region
flt = f_rgns eq 0
flt(0,0) = 0
l_pix = where(hls)
; Create a matrix for approximating intensities in these regions
mtrx = crt_mtrx(nimg, hls, 3, y_sol)
; Total number of pixels not fit to yet
;nhls = max(hl_rgns)
tn_pix = n_elements(l_pix)
; Find number of regions fit to
nrgns = max(f_rgns)
; Array of approximated values for holes
tst_rgns = fltarr(tn_pix, nrgns)
tst_rgns(*) = 99
flg_rgns = bytarr(tn_pix, nrgns)
bst_fit = fltarr(tn_pix)

; Find regions neighboring the holes
for rg = 1, nrgns do begin
    chl = hls
    tmp_rgn = f_rgns eq rg
    repeat begin
        ;tvsc1, tmp_rgn, 4
        ;tvsc1, chl, 5
        neigh_rgn = dilate(tmp_rgn, vnm)*chl
        tmp_rgn = tmp_rgn + neigh_rgn
        flg_rgns(*,rg-1) = flg_rgns(*,rg-1) + (neigh_rgn)(l_pix)
        chl = chl - neigh_rgn
        print, total(neigh_rgn)
    endrep until total(neigh_rgn) eq 0
    rcn_img(l_pix) = flg_rgns(*, rg-1)*rg
    tst_rgns(*, rg-1) = f_coef(*,rg)#mtrx
    ;tvsc1, rcn_img eq rg, 6
endfor

; Find the best approximation for each region

```

```

p = 0

while p lt tn_pix-3 do begin
  repeat begin
    p = p + 1
    posb = where(flg_rgns(p,*))
  endrep until posb(0) ne -1
  bst_fit(p) = min(abs((tst_rgns(p, *))(posb)-y_sol(p)), rg)
  fin_rgns(l_pix(p)) = (posb)(rg) + 1
endwhile
rcn_img(l_pix) = tst_rgns(indgen(tn_pix), fin_rgns(l_pix)-1)

; Find pixels that were not assigned to a region
ndef = where(fin_rgns eq 0)
rcn_img(ndef) = in_img(ndef)
return, rcn_img
end

```