

**PENGELASAN KECACATAN KIMPALAN  
MENGUNAKAN PENGENALAN JEJARI KUTUB DAN  
RANGKAIAN SARAF**

*(WELD DEFECT CLASSIFICATION USING POLAR RADIUS  
SIGNATURE AND NEURAL NETWORK)*

Oleh  
TEOW SOO PEI  
65801

Penyelia  
ASSOCIATE PROF. MANI MARAN RATNAM

Mac 2005

Disertasi ini dikemukakan kepada  
Universiti Sains Malaysia  
Sebagai memenuhi sebahagian daripada syarat untuk pengijazahan dengan kepujian  
**SARJANA MUDA KEJURUTERAAN MEKANIK**



Pusat Pengajian Kejuruteraan Mekanik  
Kampus Kejuruteraan  
Universiti Sains Malaysia

# TABLE OF CONTENTS

	Page
Table of Contents	i
List of Tables	iii
List of Figures	iv
Abstract	vi
 <b>CHAPTER 1</b>	
<b>INTRODUCTION</b>	
1.1 Background	1
1.2 Objective	2
1.3 Thesis Outline	3
 <b>CHAPTER 2</b>	
<b>LITERATURE REVIEW</b>	
2.1 Introduction	4
2.2 Overview of Automatic Defect Classification	4
2.2.1 Defect Recognition Using Shape Features	4
2.2.2 Weld Extraction	5
2.2.3 Background Subtraction Method	5
2.3 Challenges in the Development of Automated Defect Recognition	6
2.4 Basic Concept in Image Processing	6
2.5 Segmentation	7
2.5.1 Thresholding	7
2.5.2 Edge Detection	7
2.6 Blob Analysis	8
2.7 Roughness Parameter	9
2.8 Introduction of Software	11
2.8.1 <i>Matrox Inspector 2.1</i>	11
2.8.2 <i>MIL 4.0 (Matrox Imaging Library)</i>	12
2.8.3 <i>Microsoft Visual C++ 6.0</i>	12
 <b>CHAPTER 3</b>	
<b>DEFECT CLASSIFICATION USING POLAR RADIUS SIGNATURE</b>	
3.1 Introduction	13
3.2 Image Analyzing	13
3.3 Pixel	13
3.4 Methodology	14
3.4.1 Binarization	17
3.4.2 Centroid Distance and Boundary Angles	18

3.5	Results and Discussion	20
3.5.1	Problem in Polar Radius Signature	26
3.6	Conclusion	27

## **CHAPTER 4**

### **GRAPHICAL ANALYSIS USING ROUGHNESS PARAMETER**

4.1	Introduction	28
4.2	Normalization	28
4.3	Methodology	29
4.4	Results and Discussion	33
4.5	Conclusion	38

## **CHAPTER 5**

### **CLASSIFICATION OF WELD DEFECTS USING NEURAL NETWORK**

5.1	Introduction	39
5.2	Methodology	39
5.3	Results and Discussion	40
5.3.1	Classification Using Five Parameters on Four Types of Defect	42
5.3.2	Classification Using Four Parameters on Four Types of Defect	44
5.3.3	Classification Using Two Parameters on Four Types of Defect	46
5.3.4	Classification Using Five Parameters on Two Types of Defect	48
5.3.5	Classification Using Two Parameters on Two Types of Defect	50
5.4	Conclusion	51

## **CHAPTER 6**

### **CONCLUSION AND FUTURE WORK**

6.1	Conclusion and Contributions of the research	52
6.2	Recommendations and Future Work	53
6.3	Summary	54

<b>REFERENCES</b>	55
-------------------	----

<b>APPENDIX A</b>	58
-------------------	----

<b>APPENDIX B</b>	61
-------------------	----

<b>APPENDIX C</b>	66
-------------------	----

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
Table 4.1	Comparison results of $Ra$ , $Rq$ , $Rz$ for different sizes of porosity	34
Table 4.2	Distribution of five roughness parameters for four types of weld defect	34
Table 4.3	Mean value of five parameters of weld defect	37
Table 5.1	Number of real defect	40
Table 5.2	Tested data of real images	41
Table 5.3	Real image of radiograph	42
Table 5.4	Accuracy of different combination	51

## LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
Figure 2.1	Brightness histogram of the image	7
Figure 2.2	Graphical representation of Ra	9
Figure 2.3	Graphical representation of Ra and Rv	10
Figure 2.4	Graphical representation of Rz	11
Figure 3.1	Pixel coordinates system	14
Figure 3.2	Flow Chart of polar radius signature extraction	16
Figure 3.3	Histogram of the images for (a) crack (b) inc.penetration (c) porosity (d) Slag	17
Figure 3.4	Polar coordinate system	18
Figure 3.5	Quadrant	18
Figure 3.6	Images (a)circle (b) ellipse (c) rectangle (d) square (e) image1	20
Figure 3.7	Polar radius signature for (a)circle (b) ellipse (c) rectangle (d) square (e) image1	21
Figure 3.8	Simulated images for (a) crack (b) porosity (c) Slag (d) inc.penetration	23
Figure 3.9	Binary images for (a) crack (b) porosity (c) Slag (d) inc.penetration	24
Figure 3.10	Polar radius signature for (a) crack (b) porosity (c) Slag (d) inc.penetration	25
Figure 3.11	Images for (a) crack1 (b) crack2	26
Figure 3.12	Polar radius signature for (a) crack1 (b) crack2	27
Figure 4.1	Flow chart of roughness parameters extraction	31
Figure 4.2	Images of porosity	33
Figure 4.3	Distribution of roughness parameter	35

(a)  $Ra$  (b)  $Rq$  (c)  $Rz$  (d)  $Rp$  (e)  $Rv$

Figure 5.1	Three-layer neural network used to classify welded defects	40
Figure 5.2	Accuracy of classification using five parameters	42
Figure 5.3	Accuracy of classification using four parameters	45
Figure 5.4	Accuracy of classification using two parameters	46
Figure 5.5	Accuracy of classification using five parameters on two defects (crack & porosity)	48
Figure 5.6	Accuracy of classification using five parameters on two defects (crack & inc.penetration)	49
Figure 5.7	Accuracy of classification using Two parameters on two defects (crack & porosity)	50
Figure 5.8	Accuracy of classification using Two parameters on two defects (crack & inc.penetration)	50

## ABSTRACT

An automatic defect classification system was developed in this research using simulated image database and polar radius signature and neural network classifier to identify different types of defect in radiographic images of welds. Programs were developed by using *Language C* to obtain polar radius signature and roughness parameters from simulated images for subsequent classification. The image processes involved are blob analysis, binarization, edge detection, etc to extract polar radius signature. Several roughness parameters such as  $Ra$ ,  $Rq$ ,  $Rz$ ,  $Rp$  and  $Rv$  were then extracted from the polar radius signature from the simulated images. Neural network was employed to train the simulated data. A total of 4 defect types were studied and the classification was carried out using several different combinations of roughness parameters and types of weld defect. The highest accuracy of 81.25% was achieved in classifying crack and incomplete penetration by using five parameters. Therefore, roughness parameters which are extracted from polar radius signature have potential in weld defect classification.

## ABSTRAK

Satu sistem otomatis bagi pengelasan kecacatan kimpalan telah diperbangunkan dalam penyelidikan ini dengan menggunakan data daripada imej simulasi yang bersertakan pengenalan jejari kutub dan pengelas rangkaian saraf, untuk mengecam pelbagai jenis kecacatan dalam imej kimpalan radiografik. Atucara-atucara telah diperbangunkan dengan menggunakan *bahasa C* untuk mendapatkan pengenalan jejari kutub dan parameter kekasaran daripada imej simulasi untuk pengelasan seterusnya. Proses imej melibatkan analisis blob, perduaan, pengesanan pinggir dan sebagainya untuk mengekstrakkan pengenalan jejari kutub. Beberapa parameter kekasaran seperti  $Ra$ ,  $Rq$ ,  $Rz$ ,  $Rp$  dan  $Rv$  yang tersari daripada pengenalan jejari kutub daripada imej simulasi. Rangkaian saraf telah digunakan untuk melatih data simulasi. Jumlah empat jenis kecacatan kimpalan telah ditinjau dan pengelasan menggunakan beberapa kombinasi daripada jenis parameter kekasaran serta jenis kecacatan kimpalan telah dijalankan untuk tinjauan keputusannya. Kejituan paling tinggi telah dicapai dengan keputusan 81.25% daripada pengelasan retak dan penusukan tak lengkap dengan menggunakan lima jenis parameter kekasaran. Oleh itu, parameter kekasaran yang tersari daripada pengenalan jejari kutub mempunyai keupayaan dalam pengelasan kecacatan kimpalan.



# CHAPTER 1: INTRODUCTION

## 1.1 Background

Inspection of welded structures is essential to ensure that the quality of welds meets the requirements of the design and operation. A variety of non-destructive testing methods are available for the inspection of welding defects. Radiographic testing is a well-established non-destructive testing method to detect subsurface welding defects.

In this paper, an automatic computer-aided identification system was implemented to recognize different types of welding defects in radiographic images. To accomplish this research, 4 defect types which are porosity, crack, and incomplete penetration and slag inclusion were currently used for the classification. Image-processing techniques such as background subtraction and histogram thresholding were implemented to separate defects from the background. After welds are successfully extracted, polar radius signatures were generated from each welding defect. The subsequent extracted features from polar radius signature are roughness parameters are subsequently used to classify welding flaws into different types by using well-known classifiers: neural networks classifiers.

Analysis has been carried out to evaluate the effectiveness of classifying real defects using simulated training data. However, the data sets used for training and testing are limited to verify the performance of classification.

## 1.2 Objective

This project is focused on the development of automatic defect classification that will assist the inspector to interpret weld radiograph automatically. The research is carried out towards the following objectives:

1. To study and assess the development of weld defects interpretation in recent years.
2. To develop an algorithms by using *Language C* and *Matrox Imaging Library* to extract polar radius signature from the defect images for consequent graphical analysis.
3. To develop an algorithms by using *Language C* to obtain roughness parameters from the polar radius signatures.
4. To investigate the suitability of roughness parameters as feature extraction and determine the relevant roughness parameters for classification.
5. To classify the four types of weld defect by using the neural network.

### **1.3 Thesis Outline**

Sum up, this thesis has six chapters. This thesis is arranged in accordance to the objectives as mentioned above. Chapter 2 reviews the automatic defect classification in weld, which provides an overview of the related research from the early stage until the recent research of interest. Theory of relevant topics also included in this chapter.

Chapter 3 describes the image processing and preliminary weld defects classification by using polar radius signature. An algorithm is developed to extract the data from the images of weld defect which used on polar radius signature.

Chapter 4 discusses the roughness parameters and compares which parameters can distinguish the weld defects clearly. These parameters will apply in neural network for classification. An algorithm is also developed to facilitate to get the roughness parameters from each of defects.

Chapter 5 discusses the result of the accuracy from classification of weld defect. The simulated images are used as input training data; where else the real images are used as output.

Finally, conclusions of this research are drawn in Chapter 6. Recommendations and suggestions have been given for the future work.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Introduction

In this chapter, a review of automatic defect classification of weld radiograph will be made. The development of this research area will be assessed thoroughly especially in the published works on features extraction using roughness parameters for weld defect classification. This literature survey focuses on three important components, namely weld extraction, defect segmentation and specifically, classification of defect types. A special attention will be given to the recent research of interest in defect classification, where different defect types are to be characterized and classified using polar radius signature.

## 2.2 Overview of Automatic Defect Classification

The methodology used in this investigation follows the pattern recognition schema: image segmentation, feature extraction and classification. The segmentation process oriented towards the detection of edges by employing the *Laplacian edge*. This technique finds edges by looking for zero crossings. Subsequently, the *feature extraction* is centred principally on the measurement of properties of the regions. Finally, *classification* orders segmented regions in specific regions according to extracted features, assigning each region to one of a number of pre-established groups, which represent all possible types of regions expected in the image. In the detection problem, the classes that exist are only two: 'defects' or 'no defects', whereas the recognition of the type of the defects (e.g., porosity, slag, crack, incomplete penetration, etc.) is known as classification of flaws types. In this paper, some roughness parameters are studied in order to classify the defects.

### 2.2.1 Defect Recognition Using Shape Features

A method for automated recognition of welding defects was presented in (Sofia & Redouane, 2002). The detection follows a pattern recognition methodology:

- i) Segmentation: regions of pixels are found and isolated from the rest of the X-ray image using a watershed algorithm and morphological operations (erosion and dilation).
- ii) Feature extraction: the regions are measured and shape characteristics (diameter variation and main direction of inertia based on invariant moments) are quantified.
- iii) Classification: the extracted features of each region are analysed and classified using a k-nearest neighbour classifier.

According to the authors, the method is robust and achieves a good detection rate.

### **2.2.2 Weld Extraction**

In 1994 Lawson and Parker proposed in (Lawson & Parker, 1994) that artificial neural networks (ANN) be used for the automated detection of defects in X-ray images. The method generates a binary image from the test image where each pixel is either 0 when a regular structure features of the piece or 1 when a defect is detected. This entails the supervised learning of a multi-layer perceptron network (MLP) where the attempt is made to obtain detection from training data. A back propagation algorithm is used for the assignment of weightings within the MLP.

The desired detection in the training data was obtained with a segmenting procedure based on an adaptive threshold. During the experiments of five X-ray images, Lawson and Parker show that the detection using ANN is superior to the segmenting method using adapted thresholds.

The advantage of this approach is that only few samples required for the training of data.

### **2.2.3 Background Subtraction Method**

A detection approach based on a curve fitting was proposed in 1998. The key idea of this work is to simulate a 2D background of a normal welding bead characterised by low spatial frequencies in comparison with the high spatial frequencies of image of the defects. Thus, a 2D background is estimated by fitting each vertical line of the weld to a polynomial function. Thus, the obtained image is subtracted from the original image. The defects are detected where the difference is considerable. Wang & Liao (2002) and Liao (2003) propose a fuzzy  $k$ -nearest neighbour, multi-layer perceptron neural network and a fuzzy expert system for the classification welding defect types. The features used for the classification are distance from centre, circularities, compactness, major axis, width and length, elongation, Heywood diameter and average intensity and standard deviation of intensity.

In this literature review, we observe that usually features that provide information about the grey values (intensity features) are used to detect the flaws, whereas geometric features are employed to classify them.

## **2.3 Challenges in the Development of Automated Defect Recognition**

The biggest challenge in the development of automated defect recognition system is the feature identification. Radiographic images of the same defect like porosity can vary in size, shape and density. Irrespective of these variations, the software and methodology so chosen should be capable of identifying it as porosity. For effective feature identification, it is essential that at the input stage, a large number of radiographic images pertaining to that particular defect is characterised. Feature classification can be accomplished earlier using a variety of methods such as graphical analysis, pattern recognition etc.

## **2.4 Basic Concept in Image Processing**

An image defined in the "real world" is considered to be a function of two real variables, for example,  $a(x,y)$  with  $a$  as the amplitude (e.g. brightness) of the image at the *real* coordinate position  $(x,y)$ . An image may be considered to contain sub-images sometimes referred to as *regions-of-interest*, *ROIs*, or simply *regions*. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition.

In this paper, image processing used in this investigation follows the pattern recognition schema: image segmentation, feature extraction and classification.

## 2.5 Segmentation

In the analysis of the objects in images it is essential that we can distinguish between the objects of interest and "the rest." This latter group is also referred to as the background. The techniques that are used to find the objects of interest are usually referred to as *segmentation techniques* - segmenting the foreground from background. In this section we will two of the most common techniques--*Binarization* and *edge detection*.

### 2.5.1 Thresholding

In gray-scale system, typically has 4, 6, or 8 bits of memory. In this paper, 8 bits unsigned was used. In binarization, also called as thresholding; the intensity value of each pixel is ultimately reduced to either of two value, white or black, depending on whether the intensity value exceeds a given threshold level. This is done by comparing the intensity value of each pixel with a defined threshold value. If the pixel value is greater than the threshold, it is given the binary bit value of white (0); if less than defined threshold, then it is given the bit value of black (255). In most cases the threshold is chosen from the brightness histogram (shown in figure 2.1) of the region or image that we wish to segment.

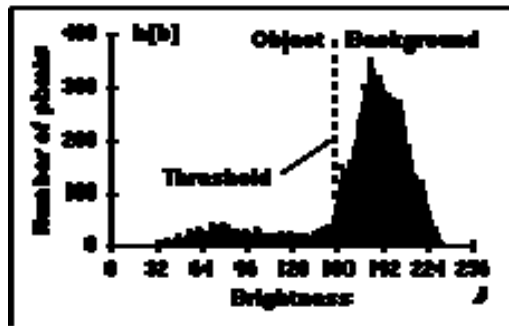


Figure 2.1: Brightness histogram of the image

### 2.5.2 Edge Detection

Thresholding produces a segmentation that yields all the pixels that, in principle, belong to the object or objects of interest in an image. An alternative to this is to find those pixels that belong to the borders of the objects. Techniques that are directed to this goal are termed *edge detection or edge finding techniques*. The edge finding technique which is used in this research is Laplacian. All of the methods based on zero crossings in the



Laplacian must be able to distinguish between zero *crossings* and zero *values*. To distinguish between these two situations, we first find the zero crossing positions and label them as "1" and all other pixels as "0". The same, in this paper, the border of the objects given the value with 255's and 0's else where.

MimConvolve(MilImage1, MilImage2, M\_LAPLACIAN\_EDGE). This is the MIL command used for edge detection. The KernelBufId parameter specifies the identifier of the kernel buffer. This parameter can be given a custom or predefined kernel identifier. The corresponding kernel in this program is *M\_LAPLACIAN\_EDGE* and their associated operation parameter is as below:

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

This 3x3 kernel is understood to mean that the central pixel brightness value is multiplied by 4, the values of 4 touching neighbors to the sides and above are multiplied by -1, and the 4 diagonally touching neighbors by 0.

## 2.6 Blob Analysis

A simple definition of a blob is a group of connected pixels. In general, blobs are thought of as group of contiguous pixels that have the same intensity. Image processing operates on these blobs to calculate the area, perimeter, or to count the number of distinguishable blobs. Before apply blob analysis, preprocess the image by converting a gray scale image (an image with 256 levels) to an image with only two gray scales – zeros and ones must be known. The objective is to separate the important objects, blobs, from the unimportant information, background. A technique called thresholding appropriately separates the blobs from the background. The result of the thresholding process is a binary image which is an image of pixel values of only ones and zeros. The blobs are represented by the connected pixels of ones, and the background is represented by the zeros. By binarizing the image into ones and zeros, the task of writing image processing algorithms for blob analysis is made easier. For example, to find the area of a blob, you simply need to count the pixels with values of one that are connected. Another benefit of binarizing the image for blob analysis is that the blob analysis calculations are fast. This application note

will introduce blob analysis with a simple cell counting application. Realize that blob analysis is used not only in biomedical imaging applications, but also in industrial inspection applications for counting components and calculating the locations of objects

In this paper, blob analysis provides application with information about the center gravity of the object.

## 2.7 Roughness Parameter

### Ra - Average Roughness

Also known as Arithmetic Average (AA), Center Line Average (CLA), Arithmetical Mean Deviation of the Profile.

The average roughness is the area between the roughness profile and its mean line, or the integral of the absolute value of the roughness profile height over the evaluation length:

$$R_a = \frac{1}{L} \int_0^L |r(x)| dx$$

When evaluated from digital data, the integral is normally approximated by a trapezoidal rule:

$$R_a = \frac{1}{N} \sum_{i=1}^N |z_i|$$

Graphically, the average roughness is the area (shown in Figure 2.2) between the roughness profile and its center line divided by the evaluation length.

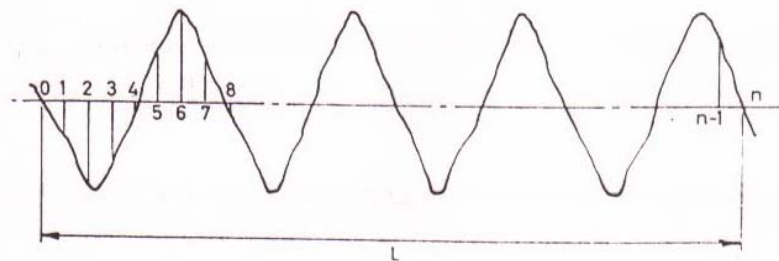


Figure 2.2: Graphical representation of Ra

## Rq - Root-Mean-Square Roughness

The root-mean-square (rms) average roughness of a surface is calculated from another integral of the roughness profile:

$$R_q = \sqrt{\frac{1}{L} \int_0^L r^2(x) dx}$$

The digital equivalent normally used is:

$$R_q = \sqrt{\frac{1}{N} \sum_{n=1}^N r_n^2}$$

Rq has now been almost completely superseded by Ra In metal machining specifications. Rq still has value in optical applications where it is more directly related to the optical quality of a surface.

## Rp, and Rv

The peak roughness Rp is the height of the highest peak in the roughness profile over the evaluation length. Similarly, Rv is the depth of the deepest valley in the roughness profile over the evaluation length.

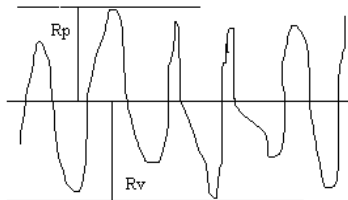


Figure 2.3: Graphical representation of Rp and Rv

## Rz

The ten-point height average (Rz) is determined by drawing a line AA parallel to the general lay of the trace, as shown in figure 2.4. In this paper, the line AA is determined at

x-axis. The height from AA to the five highest peaks and the five lowest valleys in the trace are determined. The average peak to valley height,  $Rz$ , is then given by:

$$Rz = \frac{(h1 + h3 + h5 + h7 + h9) - (h2 + h4 + h6 + h8 + h10)}{5}$$

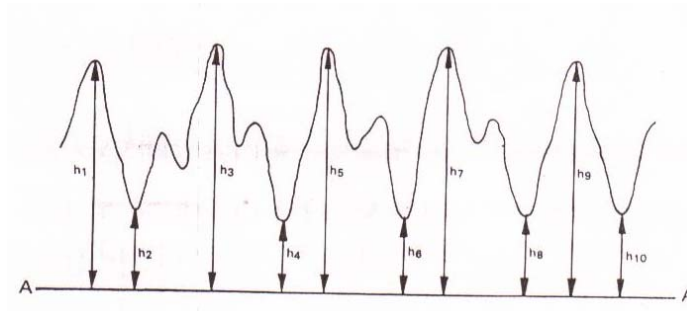


Figure 2.4: Graphical representation of Rz

## 2.8 Introduction of Software Which is Used

Several software are used to carry out this task, there are *matrox inspector 2.1*, *MIL 4.0 (Matrox Imaging Library)*, *Microsoft Visual C++ 6.0*.

### 2.8.1 *Matrox Inspector 2.1*

*Matrox Inspector 2.1* is a 32 bit application that can use in the operation of Window 95, Window NT 3.51 or NT 4.0. This software allow user for image processing such as viewing, analyzing, printing, converting, storing, modifying and etc.

*Matrox Inspector 2.1* developed with employing in *Matrox Imaging Library (MIL 4.0)*. MIL is an independent library for all the Matrox products. To carry out this objective of research, *Matrox Inspector 2.1* is used to convert RGB image to 8-bit *unsigned* image and to zoom the images to specified size. Besides that, it used to testing the accurate of program developed, which the program developed to get the center gravity and extract the data for graphical analysis.

### ***2.8.2 MIL 4.0 (Matrox Imaging Library)***

MIL is a field-proven, high-level programming library with an extensive set of optimized functions for image capture, image processing, pattern recognition, blob analysis, measurement, OCR, bar- and matrix-code recognition, calibration, graphics, image compression, image display and archiving. MIL also features the Edge Finder, a powerful tool for feature extraction and analysis. Search criteria can include or exclude features based on length, compactness, bounding box size, or interrelationship criteria, such as basic shapes that lie within other edges.

In this paper, features in MIL such as image processing, blob analysis, edge finder (Laplacian edge) are used. Blob analysis is use to determine the centroid gravity of image. Laplacian edge is use to edge detection for feature extraction and also for subsequently analysis.

### ***2.8.3 Microsoft Visual C++ 6.0***

Microsoft Visual C++ 6.0 is a fully integrated editor, compiler, and debugger. Besides that, it has ability to create complex software systems. Dennis Ritchie of Bell Labs created the C Programming Language in 1971-1972 as he and Ken Thompson worked together to design the UNIX operating system. C is hardware independent. With careful design, it is possible to write C programs that are portable to most computers.

# CHAPTER 3: DEFECT CLASSIFICATION USING POLAR RADIUS SIGNATURE

## 3.1 Introduction

This chapter focuses on the image processing and defect classification using polar radius signature. The data of images is extracted by using *Matrox Imaging Library* with *Microsoft Visual C++* software. The image processes involved are image reading, image displaying, blob analysis and a number of techniques which have been used for analyzing the image data. One category of techniques in image processing and analysis is called segmentation. Segmentation techniques are intended to define and separately regions of interest within the image. The techniques which had been used are thresholding and edge detection. These techniques proceed to carry out polar radius signature. The polar radius signature will be used for feature extraction for further analysis and classification.

## 3.2 Image Analyzing

From the early stages, the objects used for analyzing are circle, ellipse, rectangle, square and image 1 which are shown in Figure 3.5. These images were drawn by using Microsoft Paint where the object was coloured in white and the background was coloured in black. After that, *Matrox Inspector 2.1* was used to convert the images from RGB into 8-bit unsigned in tiff format for further analyzing. After that, the simulation images (crack, porosity, slag, incomplete penetration) which have been converted into 8 bit unsigned were used for getting the polar radius signature through the program developed. The simulation images are 8 bit unsigned intensity gray level images.

## 3.3 Pixel

Most commonly, the image is divided up into a rectangular grid of pixels, so that each pixel is itself a small rectangle. Once this has been done, each pixel is given a pixel

value that represents the colour of that pixel. It is assumed that the whole pixel is the same colour, and so any colour variation that did exist within the area of the pixel before the image was discretized is lost. However, if the area of each pixel is very small, then the discrete nature of the image is often not visible to the human eye.

Beside that, the most convenient method for expressing locations in an image is to use pixel coordinates. In this coordinate system, the image is treated as a grid of discrete elements, ordered from top to bottom and left to right, as illustrated by the Figure 3.1.

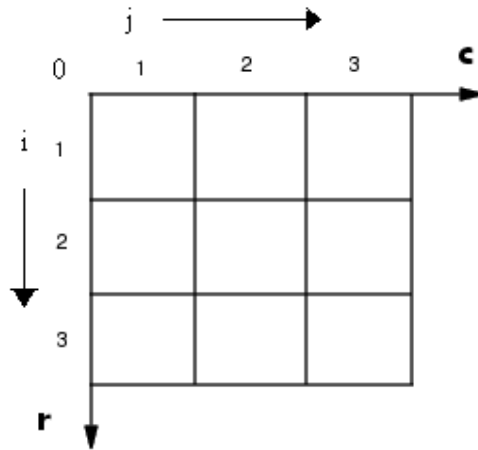


Figure 3.1: Pixel Coordinates System

For pixel coordinates, the first component  $r$  (the row) increases downward, while the second component  $c$  (the column) increases to the right. Pixel coordinates are integer values and range between 1 and the length of the row or column.

### 3.4 Methodology

In this chapter, a program (Project1a) was developed by using *Language C* (*Microsoft Visual C++*) together with *Matrox Imaging Library*. The purpose of this program is to get the polar radius signature from each type of defects for further image feature extraction. The flow chart which shows the flow of this program is shown in Figure 3.2. This image processing employed several commands in MIL. These commands synopsis are stated as below:

1. **MappAllocDefault**  
This macro sets up the requested MIL and processing environments using the defaults specified in the milsetup.h file. It can allocate and initialize a MIL

application, allocate the system to receive the MIL commands, allocate the digitizer and display, and allocate and clear a displayable image buffer on this target system, depending on what is requested.

2. MbufChild2d  
This function allocates a two-dimensional child data buffer from the specified, previously allocated parent data buffer. A child buffer inherits its type and attributes from the parent buffer.
3. MbufLoad  
Load source image into an image buffer.
4. MdispSelect  
Display the image buffer.
5. MimBinarize  
Binarize the image so that particles are represented in white and the background in black.
6. MimConvolve  
Perform a general convolution operation.
7. MbufExport  
Export a data buffer to a file.



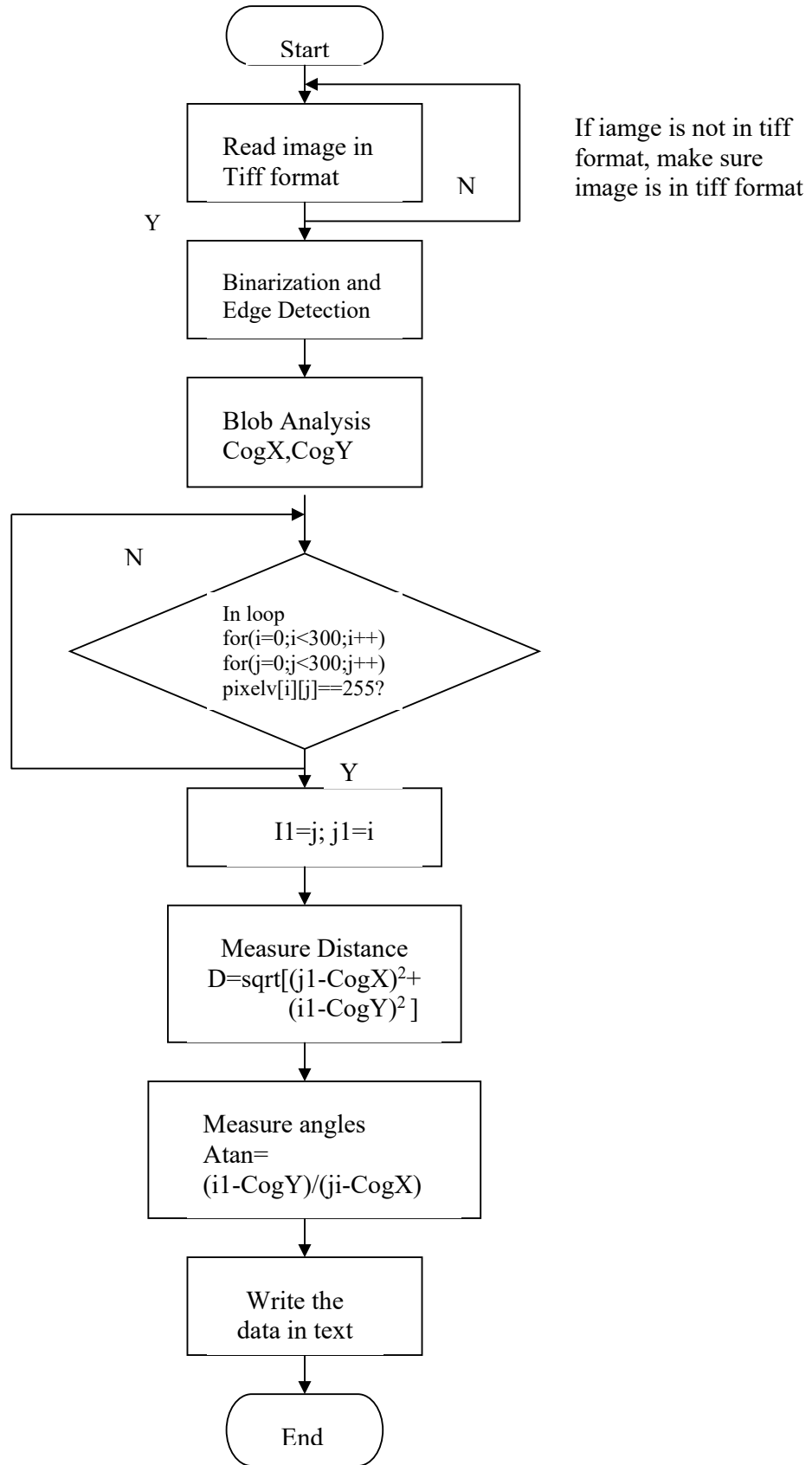


Figure 3.2: Flow Chart of polar radius signature extraction

### 3.4.1 Binarization

In the program, the threshold value of 110 was defined so that this value can commonly be used by up to four types of weld defect. This value is chosen based on the histogram of intensity for the simulated images as shown in Figure 3.3. It was found that the intensity of images of crack, incomplete penetration, porosity and slag are in the range between 128 and 78, 128 and 91, 140 and 91, 150 and 91, respectively.

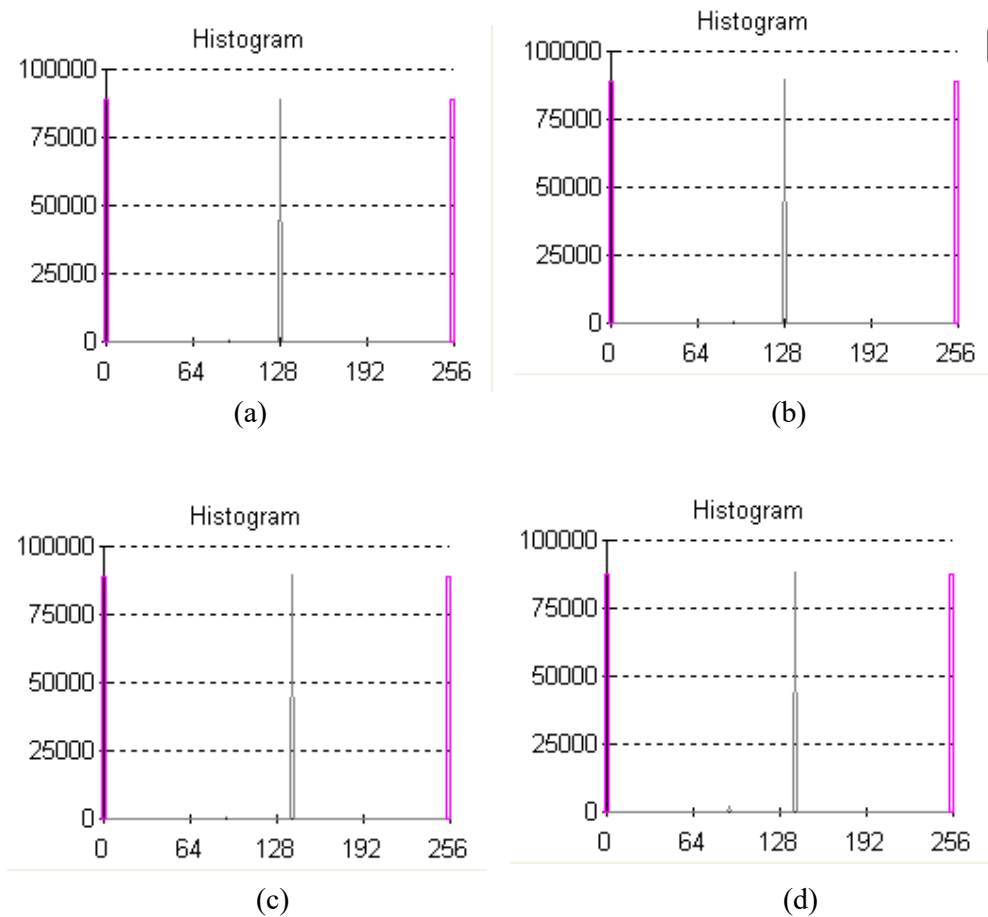


Figure 3.3: histogram of the images for (a) Crack (b) Incomplete penetration  
(c) Porosity (d) Slag

### 3.4.2 Centroid Distance and Boundary Angles

The polar radius signature involves the distance from the centroid gravity and an boundary angle. These data were extracted from the image data. The points of the boundary which value is 255 were traced to get the distance,  $r$  and the angle on that point also. Figure 3.4 shows the polar coordinate system, where  $r$  is the distance from the centroid gravity, point  $O$  to point  $P$  or points of the boundary, and  $\theta$  is the boundaries angle measured from the horizontal axis in the counterclockwise direction. The distance was measured by employing the below command in the program;

$$D=(\text{float}) \text{sqrt}(\text{pow}((\text{CogX}-i),2)+\text{pow}((j1-\text{CogY}),2));$$

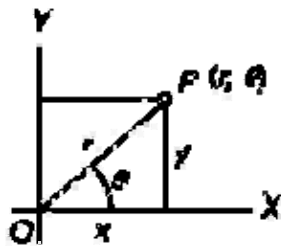


Figure 3.4: polar coordinate system

Where else, the equation which is used to measure boundaries angle is defined by each different quadrant. The first quadrant is from  $0^\circ$  to  $90^\circ$ ; the second quadrant is from  $90^\circ$  from  $180^\circ$ ; the third quadrant is from  $180^\circ$  to  $270^\circ$ ; and the forth quadrant is from  $270^\circ$  to  $360^\circ$  as shown in Figure 3.5.

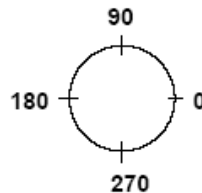


Figure 3.5: quadrant

For the first quadrant which  $i$  is less than  $\text{CogY}$  and  $j$  is greater than  $\text{CogX}$ , the angle is;

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

And the commands used in the program is

```
g=(float)(CogY-j1)/(float)(i1-CogX);  which y = (CogY-j1), x = (i1-CogX)
G=(float)atan(g); (radian)
g1=(float)(G*180/PI);
```

For the second quadrant which i and j are less than CogY and CogX, the angle is;

$$\theta = \pi - \tan^{-1} \left( \frac{y}{x} \right)$$

And the commands used in the program is

```
g=(float)(CogY-j1)/(float)(CogX-i1);
G=(float)atan(g); (radian)
g1=(float)(180-(G*180/PI));
```

For the third quadrant which i is greater than CogY and j is less than CogX, the angle is;

$$\theta = \pi + \tan^{-1} \left( \frac{y}{x} \right)$$

The commands used in the program is

```
g=(float)(j1-CogY)/(float)(CogX-i1);
G=(float)atan(g); (radian)
g1=(float)(180+(G*180/PI));
```

For the forth quadrant which i and j are greater than CogY and CogX, the angle is;

$$\theta = 2\pi - \tan^{-1} \left( \frac{y}{x} \right)$$

And the commands used in the program is

```
D=(float) sqrt(pow((i1-CogX),2)+pow((j1-CogY),2));
g=(float)(j1-CogY)/(float)(i1-CogX);
G=(float)atan(g); (radian)
g1=(float)(360-(G*180/PI));
```

### 3.5 Results and Discussion

The Figures 3.6 are the images of circle, ellipse, rectangle, square and image1 which are used in early stage to get the polar radius signature.

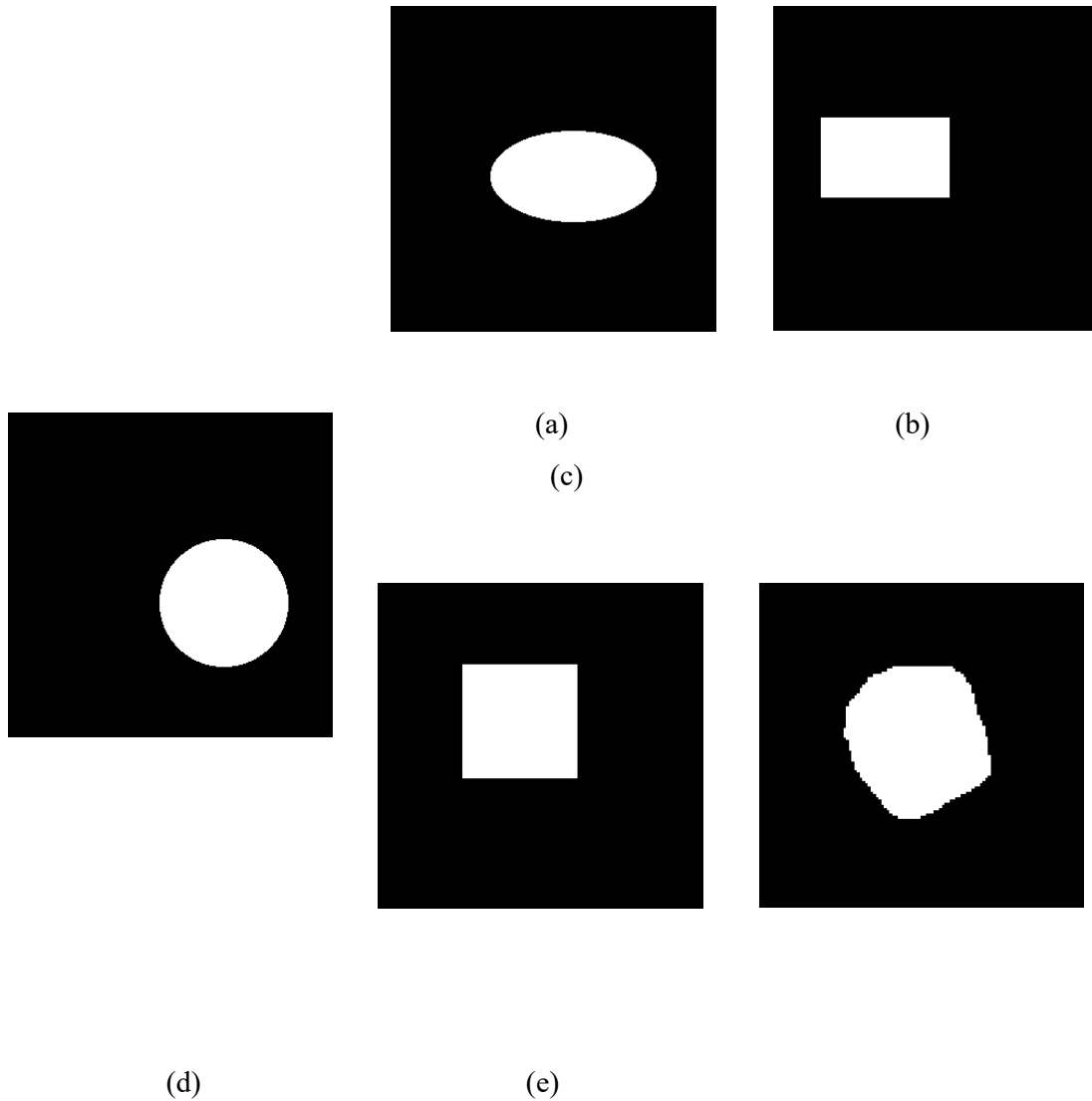
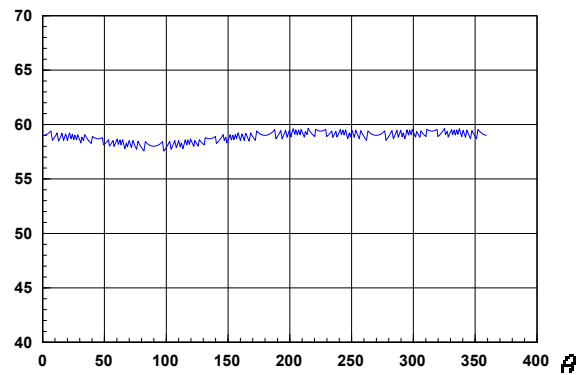


Figure 3.6: images (a) circle (b) ellipse (c) rectangle (d) square (e) image1

Figure 3.7 shows the polar radius signature extracted from the images as shown in Figure 3.6. For different shape of images, the polar radius signature is different from each

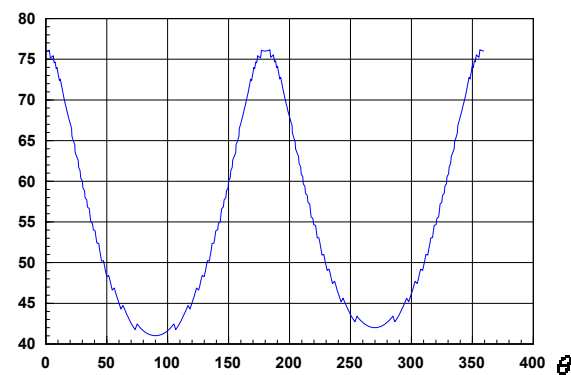
other. Therefore from the polar radius signature, the different shape of image can be distinguished from each other. However, supposed the polar radius signature for the circle is constant all over the angle. In the actual value, the radius for the circle is 58. However, the polar radius signature of the circle is fluctuated a bit from the constant value, which is the radius of the circle; besides that, the curve line for the other three types of regular shape (ellipse, rectangle and square) is also not as smooth and it is fluctuated a bit from the curve line. This is due to the fact that images are divided into a matrix of picture elements, for which each element of the matrix corresponds to a single pixel in the displayed image. Besides that, the radius is measured by the distance between the point of center gravity and the edges of the objects in terms of pixel.

Radius,  $r$



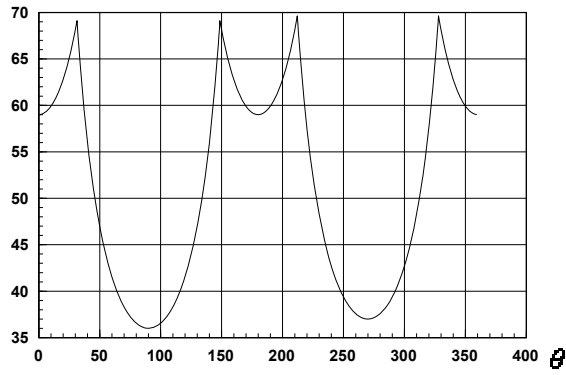
(a)

Radius,  $r$



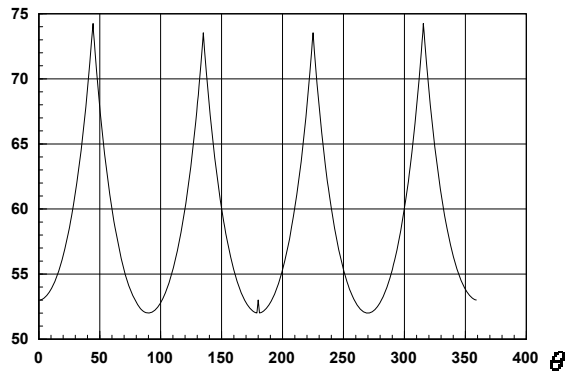
(b)

Radius,  $r$



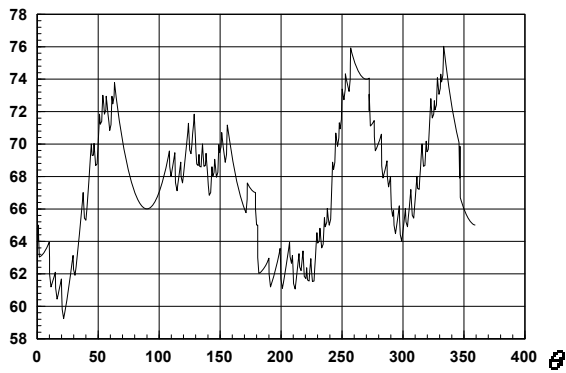
(c)

Radius, r



(d)

Radius, r



(e)

Figure 3.7: Polar radius signature for (a) circle (b) ellipse (c) rectangle  
(d) Square (e) image1

Figure 3.8 shows the four types of simulated weld defect images with image size 300x300. These simulated images were used for segmentation for subsequent polar radius signature extraction.

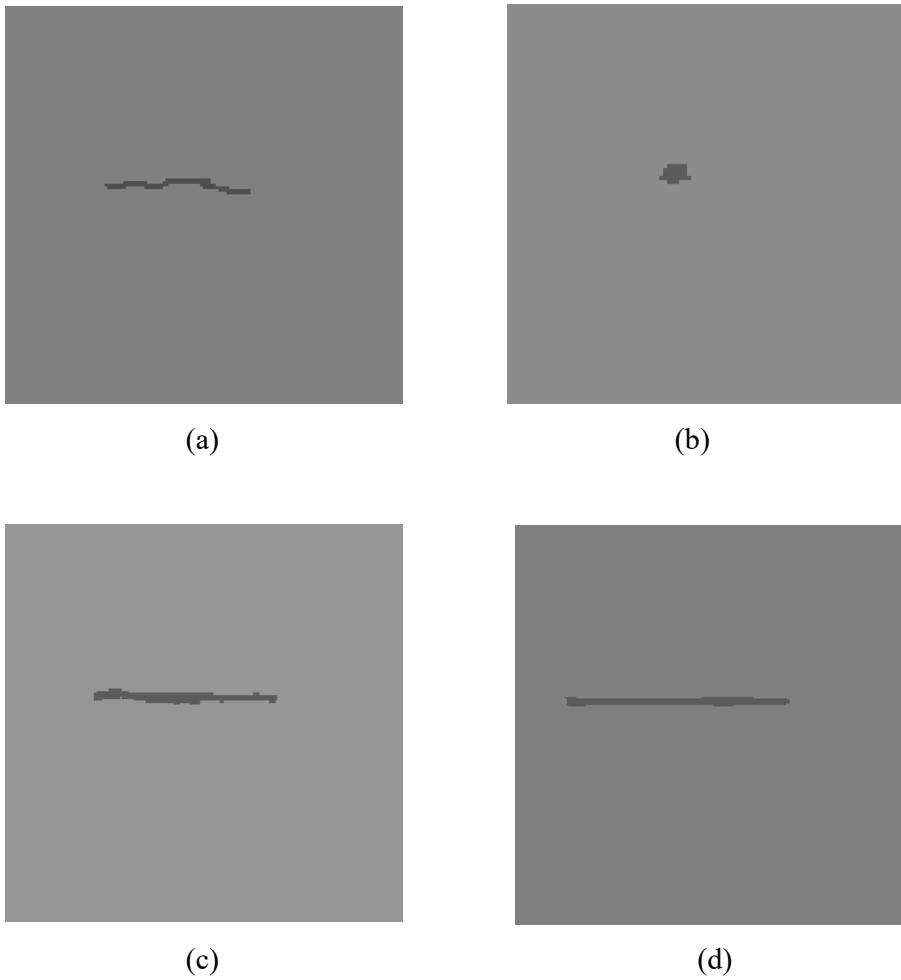
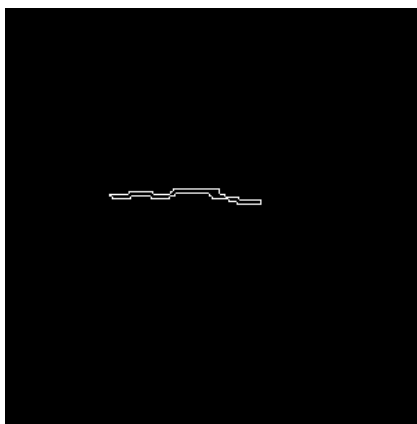


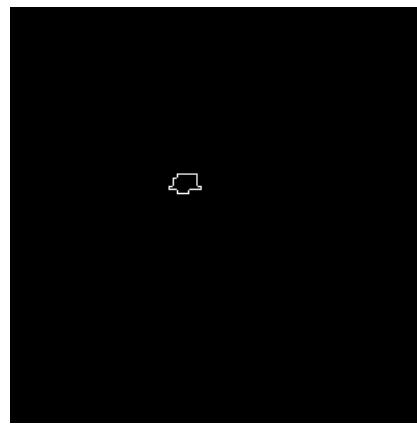
Figure 3.8: Simulated images for (a) Crack (b) Porosity  
(c) Slag (d) Incomplete penetration



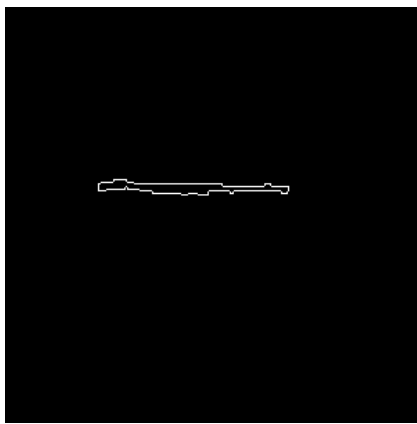
Figure 3.9 shows the images after segmentation which is an alternative to find those pixels that belong to the borders of the objects and used to extract polar radius signature.



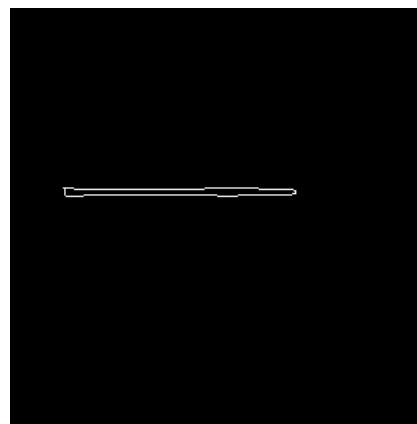
(a)



(b)



(c)



(d)

Figure 3.9: Binary images for (a) Crack (b) Porosity  
(c) Slag (d) Incomplete penetration