Surface flaw detection using digital image processing techniques.

N. Rajendran

University of Windsor

Recommended Citation
https://scholar.uwindsor.ca/etd/3677

Follow this and additional works at: https://scholar.uwindsor.ca/etd
NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE
Surface Flaw Detection Using Digital Image Processing Techniques

by

N. Rajendran

A thesis presented to the University of Windsor in partial fulfillment of the requirements for the degree of Master of Applied Science in
Department of Electrical Engineering

Windsor, Ontario, 1983

(c) N. Rajendran, 1983
ABSTRACT

An approach to detect surface flaws automatically through digital image processing techniques is investigated in this work.

The digitized image of the surface of a part to be inspected is obtained by means of a vidicon camera. The image is then properly segmented to separate the objects from the background. Thresholding can be considered as one form of segmentation. In this work some of the available thresholding schemes are investigated, and are evaluated on the basis of their accuracy and computational time.

An analysis of the objects can be made by means of their boundary points. Border-following algorithms are applied for this purpose.

Objects of interest in the image can be located by means of template matching. The use of Fast Fourier Transform (FFT), or Number Theoretic Transform (NTT) as fast means for template matching are investigated. A hardware NTT convolver is then used for the purpose of template matching.

Tools such as Display terminal, Plotter and Touch Screen are connected to the processing unit as a part of the development of a complete image processing system. Finally
the algorithms are tested on several images of industrial parts and results are presented.
ACKNOWLEDGMENTS

I would like to express my sincere thanks and gratitude to my Supervisor Dr. M. A. Siddahmed for his valuable suggestions and guidance during the course of this research. Thanks are also due to Dr. M. Shridhar for his helpful suggestions and comments on this work. A special thanks to Dr. H. K. Nanpal for his help and guidance in developing the algorithms for NTT convolver and in using the SEL computer. A special word of gratitude goes to Dr. W. C. Miller for utilizing his expertise in photography to help illustrate this work.

Last but not least, I extend my sincerest thanks to my beloved parents, without whom this work would not have been accomplished.
# TABLE OF CONTENTS

**ABSTRACT** .............................................. ii

**ACKNOWLEDGMENTS** ................................ iv

**TABLE OF CONTENTS** ................................ v

**LIST OF TABLES** .................................. viii

**LIST OF ILLUSTRATIONS** ........................... ix

<table>
<thead>
<tr>
<th>Chapter</th>
<th>INTRODUCTION</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Surface Flaws</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Digital Image Processing</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>An Image Model</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Digitized Image</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Digital Image Processing</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Quality Control using Image Processing</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Thresholding</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Border Following</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Template Matching</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Peripheral Units</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Goals of this thesis</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Thesis Organisation</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>II.</th>
<th>IMAGE THRESHOLDING</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Introduction</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Scatter-Plot Technique</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Scatter-Plot Technique</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Histogram of points having low edge values</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Histogram of points having high edge-values</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Implementation and testing of Scatter-Plot technique</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Histogram Mapping Technique</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Gray Level Transformation</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Direct specification of a bimodal histogram</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Histogram Mapping [16]</td>
<td>40</td>
</tr>
</tbody>
</table>
Results of Single-level Thresholding by
Histogram Mapping ........................................... 40
Multi-Level Thresholding .................................... 45
Direct Specification of Multi-Modal Histogram 46
Results of Multi-Level Thresholding Using
Histogram Mapping ........................................... 49
Variable Thresholding ........................................ 53
Introduction .................................................... 53
Variable Thresholding Based on Local
Bimodality ..................................................... 53
Results of Variable Thresholding Scheme ............... 59
Image Thresholding Using Gaussian Fitting
Technique ....................................................... 64
Single-level Thresholding Using Gaussian
Fitting Technique ............................................. 64
Results of Gaussian Fitting Technique .................... 64
Multi-level Thresholding Using Gaussian
Fitting .......................................................... 69
Results of Multi-Modal Gaussian Fitting
Technique ...................................................... 71
An Iterative Thresholding Scheme ......................... 74
Introduction .................................................... 74
Iterative Threshold Selection ............................... 75
Results of Iterative Thresholding Scheme ............... 77
Iterative Multi-Level Threshold Selection ................ 83
Results of Multi-level Thresholding by
Iterative Scheme ............................................. 86
Conclusions on Thresholding ................................. 88
Comparison of Various Thresholding Schemes .......... 88
Conclusions .................................................... 90

III. BORDER FOLLOWING ........................................ 91

Introduction .................................................... 91
Edge Detection Using Template Matching ................. 91
Edge Detection by Scanning .................................. 95
Boundary Following in Single-Level Thresholded
Images .......................................................... 99
Boundary Following in Multi-Level Thresholded
Images ........................................................ 102

IV. TEMPLATE MATCHING ........................................ 106

Introduction .................................................... 106
Template Matching As Applied In The System .......... 109
Direct Template Matching .................................. 109
Template Matching Using Fast Fourier Transform .... 111
Template Matching Using Number Theoretic
Transform ..................................................... 114
Template Matching Using Hardware NTT
Convolver ..................................................... 116
Results of Template Matching Using NTT
Convolver ..................................................... 117
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>V. SYSTEM INTEGRATION</td>
<td>122</td>
</tr>
<tr>
<td>Introduction</td>
<td>122</td>
</tr>
<tr>
<td>Touch Screen</td>
<td>122</td>
</tr>
<tr>
<td>Display Unit</td>
<td>125</td>
</tr>
<tr>
<td>Convolver</td>
<td>125</td>
</tr>
<tr>
<td>Problem Testing</td>
<td>126</td>
</tr>
<tr>
<td>Introduction</td>
<td>126</td>
</tr>
<tr>
<td>Testing of Piston Head</td>
<td>128</td>
</tr>
<tr>
<td>Testing of Gear Wheel</td>
<td>132</td>
</tr>
<tr>
<td>Testing of a Strain Gauge</td>
<td>136</td>
</tr>
<tr>
<td>VI. SUMMARY AND CONCLUSIONS</td>
<td>141</td>
</tr>
<tr>
<td>Summary of the work done</td>
<td>141</td>
</tr>
<tr>
<td>Conclusions</td>
<td>143</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>144</td>
</tr>
<tr>
<td>OTHER REFERENCES</td>
<td>146</td>
</tr>
<tr>
<td>VITA AUCTORIS</td>
<td>147</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table no.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison of modified histogram using low and high edge-values</td>
<td>26</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison of Edge Operators LAP, ROB and DIF</td>
<td>30</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of Various Thresholding Schemes</td>
<td>89</td>
</tr>
<tr>
<td>3.1</td>
<td>Co-ordinates of eight neighbours</td>
<td>100</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure no.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Block Diagram of an Image Processing System for Surface Flaw Detection</td>
<td>5</td>
</tr>
<tr>
<td>1.2a</td>
<td>A Sample Histogram Illustrating a Bi-modal Distribution</td>
<td>8</td>
</tr>
<tr>
<td>1.2b</td>
<td>A Sample Histogram Illustrating a Multi-modal Distribution</td>
<td>8</td>
</tr>
<tr>
<td>1.3a</td>
<td>A Point Template</td>
<td>11</td>
</tr>
<tr>
<td>1.3b</td>
<td>A Sample Image to Illustrate Template Matching</td>
<td>11</td>
</tr>
<tr>
<td>2.1a</td>
<td>A Sample Image Illustrating an almost Uni-modal Distribution</td>
<td>21</td>
</tr>
<tr>
<td>2.1b</td>
<td>Histogram of Sample Image in Fig. 2.1d</td>
<td>21</td>
</tr>
<tr>
<td>2.2a</td>
<td>Original image of the surface of a Strain Gauge</td>
<td>27</td>
</tr>
<tr>
<td>2.2b</td>
<td>Thresholded Image of Fig. 2.2a Using Low Edge-Value Points</td>
<td>27</td>
</tr>
<tr>
<td>2.2c</td>
<td>Thresholded Image of Fig. 2.2a Using High Edge-Value Points</td>
<td>28</td>
</tr>
<tr>
<td>2.3a</td>
<td>Original Image of Top Surface of Piston Head</td>
<td>31</td>
</tr>
<tr>
<td>2.3b</td>
<td>Thresholded Image of Fig. 2.3a Using Edge Operator ROB</td>
<td>31</td>
</tr>
<tr>
<td>2.3c</td>
<td>Thresholded Image of Fig. 2.3a Using Edge Operator DIF</td>
<td>33</td>
</tr>
<tr>
<td>2.3d</td>
<td>Thresholded Image of Fig. 2.3a Using Edge</td>
<td></td>
</tr>
</tbody>
</table>
2.9f Thresholded Image Obtained Using Specified Histogram in Fig. 2.9e ........................................ 52
2.10a Segmentation of Image for Variable Thresholding ......................................................... 54
2.10b Threshold Computation for Each Pixel .......... 54
2.11a Original Image or Surface of Piston Head ....... 61
2.11b Thresholded Image of Fig. 2.11a Using Variable Thresholding ........................................ 61
2.12a Original Image of Three Wooden Blocks ......... 62
2.12b Thresholded Image of Fig. 2.12a Using Variable Thresholding ........................................ 62
2.13a Original Image of Surface of a Piston Head .... 63
2.13b Thresholded Image of Fig. 2.13a Using Variable Thresholding ........................................ 63
2.14a Original Image of Surface of a Piston Head .... 66
2.14b Thresholded Image of Fig. 2.14a Using Gaussian Fitting Technique ................................. 66
2.15a Original Image of a Strain Gauge ................. 67
2.15b Thresholded Image of Fig. 2.15a Using Gaussian Fitting Technique ................................. 67
2.16a Original Image of another Strain Gauge ......... 68
2.16b Thresholded Image of Fig. 2.16a Using Gaussian Fitting Technique ................................. 68
2.16c Histogram of the Image Shown in Fig. 2.16a .... 72
2.16d Multi-modal Gaussian Fitting for Image Shown in Fig. 2.16a ......................................... 72
2.16e Multi-level Thresholded Image of Fig. 2.16a
Using Gaussian Fitting Technique .......... 73

2.17a Initial Estimation of Thresholded Image .... 75

2.17b Schematic Diagram for the Iterative Processor . 75

2.18a Original Image of Surface of a Piston head .... 78

2.18b Thresholded Image of Fig. 2.18a Using Iterative Technique ........................................ 78

2.19a Original Image of a Strain Gauge ................... 79

2.19b Thresholded Image of Fig. 2.19a Using Iterative Technique .......................................... 79

2.20a Original Image of Surface of a Piston Head .... 80

2.20b Histogram of the Image Shown in Fig. 2.20a .... 80

2.20c Thresholded Image of Fig. 2.20a Using Iterative Technique ........................................... 81

2.21a Original Image of a Section of a Gear Wheel ... 82

2.21b Thresholded Image of Fig. 2.21a Using Iterative Technique .......................................... 82

2.22a Initial Estimation of a Multi-level Thresholded Image ...................................................... 85

2.22b Schematic Diagram of Iterative Technique for Multi-level Thresholding .............................. 85

2.23 Multi-level Thresholded Image of Fig. 2.20a Using Modified Iterative Technique ................... 87

2.24 Multi-level Thresholded Image of Fig. 2.21a Using Modified Iterative Technique ................... 87

3.1 A Sample 3 by 3 Image Region for Sobel's Gradient ............................................................. 93

3.2a A Horizontal Edge-detection Template ............ 93
3.2b A Vertical Edge-detection Template .......... 93
3.3a Thresholded Image of a Piston Head .......... 94
3.3b Edge-detection of Fig. 3.3a Using Sobel's Gradient ................................................. 94
3.4a A Sample Thresholded Image .................. 96
3.4b Edge-detection in Fig. 3.4a by Scanning .... 98
3.5a Thresholded Image of a Gear Wheel .......... 98
3.5b Edge-detected Image of Fig. 3.5a by Scanning 98
3.6a Thresholded Image of a Piston Head .......... 101
3.6b Border followed Image of Fig. 3.6a .......... 101
3.7 Border followed Image of Fig. 3.5a .......... 104
3.8a Multi-level Thresholded Image of a Piston Head 105
3.8b Border followed Image of Fig. 3.8a .......... 105
4.1 A Sample Template ................................ 108
4.2a A Vertical Line Template ........................ 108
4.2b A Horizontal Line Template .................... 108
4.2c An Angled Line Template (−45 deg.) .......... 108
4.2d An Angled Line Template (+45 deg.) .......... 108
4.3a A Sample Border followed Image ............... 110
4.3b A Vertical Line Template ........................ 110
4.4a Border followed Image of a Piston Head ...... 119
4.4b Template Matched Image of Fig. 4.4a .......... 119
4.5a Border followed Image of a Gear Wheel ...... 120
4.5b Template Matched Image of Fig. 4.5a .......... 120
4.6a Horizontal and Vertical Line Template .......... 121
4.6b Template Matched Image of Fig. 4.6a Using Template Shown in Fig. 4.6a .......... 121
5.1  A Sample Menu Pad ............................................ 124
5.2  Flow of the Algorithm ........................................ 127
5.3a  Displayed Results for a Piston Head ...................... 130
5.3b  A Sample Report of the Results for Piston Head .......... 130
5.4a  Results of another Piston Head .......................... 131
5.4b  Report of the Results Shown in Fig. 5.4a .............. 131
5.5a  A Differential Gear Wheel .................................. 133
5.5b  Digitized Image of a Section of a Gear Wheel .......... 133
5.6a  Displayed Results for the Gear Wheel ................... 135
5.6b  A Sample Report of the Results of Gear Wheel .......... 135
5.7  Image of a Strain Gauge ................................. 138
5.8a  Displayed Results for a Strain Gauge .................... 139
5.8b  A Sample Report of the Results of Strain Gauge ........ 139
5.9a  Displayed Results for another Strain Gauge ............ 140
5.9b  Report of Results Shown in Fig. 5.9a .................... 140
Chapter I

INTRODUCTION

1.1 INTRODUCTION

Investigations into scientific means to obtain quality assembly line products started at the beginning of the twentieth century. It was analyzed and found that the work performance of a worker can be judged by two major parameters [1], which are: his production capacity and the quality of his production. Based on this, investigations into means of improving these two parameters were carried out. Extensive research in this area led into developing new fields, such as, industrial environment, industrial psychology, quality control etc.

In the industrial environment area, studies were conducted on the layout of the industry, its lighting conditions, its ventilation etc. A lot of importance was given for the working conditions of the employee, in order to improve his capacity or production.

In the field of quality control, scientific means were introduced to judge the quality of products. In 1924 Shewhart of Bell Laboratories applied a quality control chart on the parts supplied to the laboratory [2,3]. Since then this field has been ever expanding and new dimensions
have been added to it. Areas such as Automatic Quality Control were developed, and employed to facilitate a high-throughput rate of quality products.

In this modern age of digital computers a lot of research is being performed in the area of automatic quality inspection. It is possible to use a computer vision system to perform inspection over the manufactured or supplied parts and to detect any type of defect.

1.2 SURFACE FLAWS

A part can be defective in many ways. Concentrating more on machine parts, defects such as cracks, pores or scratches can occur in the manufacturing process. The thrust of this work will be directed to these surface flaws.

Digital image processing techniques such as thresholding, border-following, template matching etc are applied for solving this problem.

1.3 DIGITAL IMAGE PROCESSING

1.3.1 An Image Model

An image, in the general sense, can be considered as a representation of an object, as for example, a photograph. In this a scene is represented by means of various shades through optical means. The three dimensional (spatial coordinates) scene is represented only by two dimensions. However a third dimension, viz, the shade or gray level of the image is also used in representing the object.
1.3.2 **Digitized Image**

To process any given image by a digital computer it is necessary to sample and discretize the gray levels of the image. The image is then represented as a two-dimensional matrix whose elements represent the gray levels and their relative position in the image. This can be achieved for example, by means of a vidicon camera and analog to digital converter. The two-dimensional matrix is then stored in the computer memory for analysis.

1.3.3 **Digital Image Processing**

The two dimensional array of discrete values can now be used for several purposes. In some cases it might be necessary to initially enhance the image through filtering. The image is then presented for analysis. The analysis is carried-out through separating the gray levels into two or more regions representing various features and background in the image. This is then followed by border-following and/or template matching to extract various features. Such operations are generally grouped under the term digital Image Processing.
1.4 Quality Control Using Image Processing

As referred to in the title, this work is aimed at developing an image processing scheme that can be used for the purpose of surface flaw detection. The proposed scheme can be outlined with the help of the block diagram shown in Figure 1.1.

As shown in the block diagram, the image of the part to be analysed can be obtained by means of a Vidicon camera. This signal can now be discretized by means of an A/D Converter. The digitized image can then be passed through a pre-processing procedure to remove noise, reflections etc and then stored in the memory of a digital computer as a two dimensional array.

The digital image after pre-processing can now be used for various analyses. The image on the whole contains a lot of information. However it is necessary to undergo certain steps to use this information content. The image might contain more than one object in its range. Again all these objects would be grouped under a common background. Hence it is necessary to separate all these objects from their background. Thus some kind of segmentation algorithm is required for this purpose. Thresholding is one fast means of segmenting the image. This step can be used to separate the objects from the background. The objects thus separated can now be analysed for any information content. However it is required to have the information about the location,
FIG. 1.1 Block Diagram of an Image Processing System for Surface Flaw Detection
size, boundary etc of these regions. Hence it is necessary to obtain the borders of the various regions in the image. Even though the borders of the regions yield useful information about the region yet in some cases it might be required to pin-point the location of certain features. The location of such inherent features is made possible by means of template matching. By using proper templates it is possible to locate the inherent features in the image. Thus the image undergoes several image processing algorithms and the information content can be effectively extracted. A brief introduction to these various algorithms is given below.

1.5 **THRESHOLDING**

The quantized gray levels in the digitized image may lie anywhere in the range of, say, 0 to 255. However in most of the cases not all the levels are fully utilized in defining the image. This will be more obvious when the probability distribution, of the gray levels of the image, is plotted. The plot of the probability distribution is often termed as the histogram of the image. By looking at the histogram of an image it is seen that more often they are clustered into two distinct groups as shown in Figure 1.2a. These groups normally represent the two populations of the image, viz, the object and the background. This immediately gives us the idea that the entire gray level
range of the image can be reduced to two distinct levels. By declaring the levels below the level $T$, shown in the Figure 1.2a, as 0 and the levels above $T$ as 255 it is possible to obtain a binary level image which is easier to work with. However there might be cases where the image might contain more than two distinct populations, as shown in Figure 1.2b. In these cases it is then required to group the levels into more than two values. This type of segmentation of a given image into different regions is called Thresholding. In the case of Figure 1.2a it is called Single-level thresholding and the value $T$ is known as the the Threshold Value of the image. However for cases shown in Figure 1.2b it is called multi-level thresholding with threshold values $T_1$ and $T_2$. Now it is required to find some means to obtain these value(s) automatically, for any given image.
FIG. 1.2a A Sample Histogram Illustrating
A Bi-modal Distribution

FIG. 1.2b A Sample Histogram Illustrating
A Multi-modal Distribution
1.6 **BORDER FOLLOWING**

A segmented image can be obtained through proper thresholding. However in most of the cases the regions as such cannot be used for analysis. The boundaries between these various regions can be obtained in several ways. By passing the segmented image through a high pass filter it is possible to extract the sharp transitions between regions and hence the boundaries. Also by using different types of edge templates it is possible to locate the boundaries. However, for a proper analysis it is necessary to isolate the co-ordinates of the border points of each region. Hence it is required to have a border following algorithm. Once a border point is detected the entire contour of that region can be followed. In this manner it is possible to obtain all the co-ordinates of the boundary of the different regions.

1.7 **TEMPLATE MATCHING**

Template Matching is one form that can be used to detect some invariant regional property present in an image. As an example, consider the template shown in Figure 1.3a. The center of the template (marked d) is moved around the image from pixel to pixel. At every position, we multiply every point of the image that is inside the template by the number indicated in the corresponding entry
of the template, and then add the results. If all image points inside the template area have same values (constant background), the sum will be a zero. If on the other hand, the center of the template is located at an isolated point the sum will be different from zero. For example, if the template is moved over the sample image shown in Figure 1.3b a maximum response is obtained at the isolated points (marked 9). By using a threshold, the weaker responses can be eliminated and a particle is detected when the sum exceeds the threshold value, its location being at the center of the template.

Such template matching techniques can be employed to detect some of the inherent features in the image.
### FIG. 1.3a A Point Template

```
<table>
<thead>
<tr>
<th>-1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>8</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
```

### FIG. 1.3b A Sample Image to Illustrate Template Matching

\[
\begin{array}{cccccccccccc}
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 9 & 3 & 3 \\
3 & 3 & 9 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 9 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 9 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
\end{array}
\]
1.8 **PERIPHERAL UNITS**

As shown in the block diagram in Figure 1.1 several peripheral units are connected to the processor in the system. The first important peripheral unit is the digital filter that is required for filtering the images as a part of pre-processing. It shall be seen that the filtering operation and template matching operation are very much similar. Thus if a hardware convolver is available it can be used for both purposes, viz, filtering as well as template matching. Thus the two blocks in the diagram can in fact be grouped into one.

When dealing with problems in image processing, it is required to see the input, output, and also the intermediate results of the various algorithms. Hence it is necessary to have a display unit attached to the system.

In some cases it might be desirable to obtain a hard copy of the results of the algorithms. Hence it is necessary to have some form plotter that can plot the results.

It would be convenient to have a general algorithm that can work on all the problems that are to be tested. Especially in surface flaw detection it is necessary to have a general algorithm that can analyze a wide range of parts. Such an algorithm, if not impossible, is yet too complicated in nature. Some parts may need some extra steps to analyze
them. Hence it would be efficient to have some means to indicate the part that is to be analysed. Some form of switch can be used for this purpose.

The touch screen shown in the block diagram can be utilised for this purpose. This feature is especially suitable for industrial environments where the switching has to be done by a shop-floor worker. A menu pad displaying the various parts that can be inspected, is displayed on the screen. The worker can then be allowed to select the appropriate part by merely touching the screen. Thus the touch screen can be utilised as a switch for indicating the part to be inspected.

Apart from the above peripheral units, the rest, viz., the storage, and console terminal are some of the regular features in any digital system which do not require a detailed discussion.

1.9 GOALS OF THIS THESIS

The goals that are set forth in this thesis can be grouped as follows:

1. Development of software for a surface flaw detection system using Digital Image Processing techniques.

2. Development of driver routines for a hardware MIT convolver.

3. Development of basic routines for using various peripheral units.
The above goals can be expanded in terms of software requirements as follows:

1. Software for automatic thresholding of a given image.
2. Software for border following on a thresholded (both single and multi-level thresholded) image.
3. Software for Template Matching to locate inherent features in an image.
4. Driver routines for the hardware NTI convolver.
5. Software for filtering given images using the NTI convolver.
6. Basic software for displaying images on the graphics terminal.
7. Basic software for plotting the images on a printer.
8. Driver routines for using the PROM programmer.

In cases of thresholding and border following a lot of work has already been done. However in case of thresholding it is necessary to select a thresholding technique that would be suitable for the purpose of surface flaw detection. Again in the case of border following it is necessary to extend the existing technique to follow borders even in multi-level thresholded images.

Hence it is required to study the above problems in a detailed manner to come up with an optimum result.
1.10 **THESIS ORGANISATION**

Chapter II gives a detailed account on image thresholding. Several of the techniques available in the literature, have been selected and tested. Improvements are made on these algorithms and where possible, extension to multi-level thresholding has been carried out.

Chapter III contains a brief discussion on some of the border following algorithms. A brief description of how the algorithm works on single-level thresholded images and also how it has been extended to locate borders on multi-level thresholded images is given. Results are shown for both single and multi-level thresholded images and it is illustrated how they are useful in template matching algorithms.

Chapter IV gives us a detailed discussion on Template Matching in Digital Image Processing. Several approaches have been considered for this purpose and are critically evaluated. Examples are given to illustrate the use of template matching in locating inherent features.

Chapter V gives a general overview of the entire system that will be required to detect surface flaws. A brief discussion on how the various algorithms and peripherals are linked together in detecting surface flaws, is given. A testing of this algorithm on three specific problems, viz, a piston head, a gear wheel and a strain gauge, is provided. Experimental results obtained through these algorithms on several images is also provided.
Chapter VI finally summarises the author's contribution in this work and discusses the conclusions derived.
Chapter II

IMAGE THRESHOLDING

2.1 INTRODUCTION

Features of interest in an image may often be extracted from their surroundings using a thresholding technique [4] in which all gray levels below the threshold are mapped black, those levels above are mapped into white, or vice-versa. The success of the technique depends on the object that is desired to be extracted, occupying a range of gray levels distinct from that of the background. In practice it is difficult to select the optimum threshold, especially if a range of image scenes with widely differing properties is being considered, and some automatic means of threshold selection is required in each case. Thresholding at too high a level results in loss of information, while thresholding at low levels can give rise to objectionable background clutter [5].

A common method of automatically deriving a threshold at which to segment a given picture is to examine its gray level histogram [6]. The presence of two peaks in such a histogram demonstrates the existence of two distinct

---

1 The histogram of an image is the plot of the probability distribution of the gray levels of the image versus gray level.
brightness regions in the image, one corresponding to the object and the other to its background. It is reasonable therefore, to choose the threshold at the gray level midway between the two peaks.

Several methods have been proposed that produce a transformed gray level histogram in which the valley is deeper, or is converted into a peak, and is thus easier to detect. The transformed histograms used in these methods can be obtained by creating a Scatter-Plot\(^2\) for the given image and computing various weighted projections of this plot onto the gray level axis. Such a method was proposed by Weszka and Rosenfeld [7]. This method was considered since it looked at a transformed histogram in selecting a threshold value. A discussion on this technique is given in the following section.

\[2.2\] SCATTER- PLOT TECHNIQUE

\[2.2.1\] Introduction

Let us assume that the given images consist of objects on a background where the objects and background each have a unimodal gray level population. Let us also assume that the gray levels of adjacent points interior to the objects, or to the background, are highly correlated, while across the edges at which object and background meet, adjacent points

\[\text{\textsuperscript{2} Scatter-Plot of an image is the plot of the edge-values, on the different gray levels of the image, against the gray levels.}\]
differ significantly in gray level.

For an image that satisfies these assumptions, its gray level will be primarily a mixture of two unimodal histograms corresponding to the object and background populations respectively. If the means of these populations are sufficiently far apart, their standard deviations sufficiently small, and they are comparable in size, the image histogram will be unimodal. Otherwise, the histogram may be unimodal, but one side may be less steep than the other, reflecting the presence of two peaks that are close together or that differ greatly in height. The histogram will also contain a third, usually smaller, population corresponding to points on the object/background border.

The presence of these border point gray levels raises the level of the valley floor between the two peaks or if the peaks are already closer together, makes it harder to detect the fact that they are not a single peak. This fact is clearly illustrated by a sample image shown in Figure 2.1a and its histogram is shown in Figure 2.1b. The object size is too small when compared to the size of the background, and also the means of the two populations are very close together. It can also be seen that the gray level population of the border points raises the floor between the two peaks thereby making it appear as a unimodal distribution. For images of this nature the Scatter-Plot technique yields a solution. Using this technique the
valley between the two peaks can be made deeper or it can be converted into a peak such that its identification becomes easier.
FIG. 2.1a A Sample Image Illustrating An Almost Uni-modal Distribution

FIG. 2.1b Histogram of Sample Image in Fig. 2.1a
2.2.2 Scatter-Plot Technique

In the Scatter-Plot technique, the rate of change of gray level at a point as well as the gray level at that point are taken into account. The rate of change of gray level is often termed as the edge-value at a point. Different types of edge-value operators are available in the literature. A few are given below.

\[
\begin{array}{cccc}
A & B & C \\
D & E & F & G \\
H & I & J & K \\
L & M
\end{array}
\]

The edge value at the point 'E' can be obtained by:

a) LAP, the Laplacian operator, defined as

\[
E - \frac{A + B + C + D + F + H + I + J}{8} \quad (2.1)
\]

b) ROB, the Roberts Cross, defined as

\[
\text{MAX} \left[ |E - C|, |B - F| \right] \quad (2.2)
\]

c) DIF, the maximum of differences of average gray level in a pair of horizontally and
vertically adjacent 2 by 2 neighbourhoods

\[
\frac{1}{4} \text{MAX} \begin{vmatrix} B & C & E & F - I & J - L - M \\ D & E & H & I - F - G - J - K \end{vmatrix} \quad (2.3)
\]

Using these operators the edge-value at a given point can be easily obtained. A plot between the gray levels and the edge-value is often termed as the Scatter-plot of the image.

2.2.2.1 Histogram of points having low edge values

According to the image model described in section 2.1, the points interior to the objects and background should generally have low edge-values (close to zero), since they are highly correlated with their neighbours, while those on the object/background border should have high edge-values. Thus if we produce a histogram of the gray levels of points having low edge-values only, the peaks should remain essentially the same, since they correspond to interior points, but the valley should become deeper, since the intermediate gray level points on the object/background border have been eliminated [8].

More generally, a weighted histogram can be computed in which points having low edge-values are counted heavily. For example if \( | \Delta | \) is the edge-value at a given point then a weight of \( 1/(1+| \Delta |^2) \) can be used. This gives a
maximum weight to points having zero edge-value and negligible weight to high edge-value points.

2.2.2 Histogram of points having high edge-values

Conversely, suppose that a histogram is plotted for gray levels of only those points that have high edge-values, then a single peak should occur at a value intermediate between the object and background gray levels. Thus the mode or perhaps the mean of the histogram should be a good threshold level \([9,10]\).

More generally a weighted histogram can be computed in which the points with high edge-values are counted more heavily. Suppose if \(|\Delta|\) was the edge-value or a point then a weight of \(|\Delta|\) should eliminate points with zero edge-values (i.e. object and background points). Watanabe \([11]\) proposed a method in which, a weight equivalent to summing the edge-values for each gray level, was used. If the edge-value at the object/background borders are very high, the resulting histogram will have a peak at the intermediate gray level, and thus the occurrence of the peak can be used as a threshold. However if the areas of the object and background are large, the sum of large numbers of low edge-values in the interior may be higher than or equal to the sum of smaller number of high edge-values at the borders, and the peak may not exist. To avoid this problem Weszka et al \([12]\) suggested that the average, rather than
the sum of the edge-values should be used as weights of the histogram.

2.2.3 Implementation and testing of Scatter-Plot technique

As seen earlier in the sequence of algorithms, the thresholding of an image was one of the important procedures in our attempt to detect surface flaws. It is also desired to implement this scheme in a real-time environment and hence it is necessary that the computational time needed for the algorithms are as low as possible. For this reason the edge operators LAP, ROB and DIF were chosen to find the edge-values in the image. As seen above the Scatter-Plot technique can be applied in several ways. We can choose to use the modified histogram of low edge-value points. This again can be sub-divided into three cases based on the edge operator used. Similarly we can have another three cases under the modified histogram using only high edge-value points. Hence it becomes necessary to find an optimum solution that yields the best results. Hence a test was conducted to select the main division i.e modified histogram. The edge operator LAP was used in both cases to compute the edge-values.

Figure 2.2a shows the original image of the surface of a strain gauge. The thresholded image obtained using the modified histogram of low edge-value points is shown in Figure 2.2b. Figure 2.2c shows the thresholded image
obtained using modified histogram of high edge value points. It is easily seen that the histogram of low edge-value points does not yield proper results when compared to the case of high edge-value points. The test was conducted on several images, and the results are tabulated in Table 2.1. From the results it is clear that the modified histogram using high edge-value points yields better results. Hence the further testing was conducted using only the histogram of high edge-value points.

**TABLE 2.1**

Comparison of modified histogram using low and high edge-values

<table>
<thead>
<tr>
<th>Image</th>
<th>Threshold using low edge-value</th>
<th>Correct Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRAIN1</td>
<td>54</td>
<td>42</td>
</tr>
<tr>
<td>STRAIN2</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>STRAIN3</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>STRAIN4</td>
<td>12</td>
<td>17</td>
</tr>
</tbody>
</table>

It is now required to select the edge operator that yields proper threshold values and is also computationally efficient. The original image of the top surface of a piston head is shown in Figure 2.3a. Figure 2.3b shows the thresholded image of Figure 2.3a obtained using high edge-
FIG. 2.2a Original Image of the Surface of a Strain Gauge

FIG. 2.2b Thresholded Image of Fig. 2.2a Using Low Edge-Value Points
FIG. 2.2c Thresholded Image of Fig. 2.2a Using High Edge-Value Points
value points with edge operator ROB. The result obtained using DIF as the edge operator is shown in Figure 2.3c and Figure 2.3d illustrates the result of edge-operator LAP. As seen from the above figures the results obtained using LAP, ROB and DIF as edge operators are very close to each other. However the computational time (given below each Figure) shows that the edge operator ROB is more efficient than the other two operators. To confirm this result the test was conducted on several number of images. The results are grouped in Table 2.2 It can be seen that the edge operator ROB is computationally efficient, even though the other two operators yield similar results. Hence it can be concluded that the modified histogram using high edge-value points with Roberts Cross (ROB) as the edge operator yields proper results with less computational time. Further comparisons (with other techniques) are made only with the results of ROB edge operator.

As seen above the Scatter-Plot technique yields good threshold values in some of the cases. However it is not the same in all the cases that are tested. Figure 2.4a shows the original image of the surface of a piston head. Figure 2.4b shows the thresholded image of Figure 2.4a using the Scatter-Plot technique. It is seen that the threshold value obtained in this case is far from correct. This is because the original image does not satisfy the initial assumptions made, viz, that the interior of the object and
### TABLE 2.2
Comparison of Edge Operators LAP, ROB and DIF

**Scatter-Plot Technique**

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>LAP</th>
<th>ROB</th>
<th>DIF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>IT</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>Sec</td>
<td></td>
<td>Sec</td>
</tr>
<tr>
<td>PSTZ</td>
<td>3.1</td>
<td>77</td>
<td>2.0</td>
</tr>
<tr>
<td>TRANS128</td>
<td>3.2</td>
<td>81</td>
<td>2.1</td>
</tr>
<tr>
<td>PHS7</td>
<td>3.2</td>
<td>98</td>
<td>2.0</td>
</tr>
<tr>
<td>PHSb</td>
<td>3.2</td>
<td>94</td>
<td>2.0</td>
</tr>
<tr>
<td>STRAIN3</td>
<td>3.2</td>
<td>20</td>
<td>2.0</td>
</tr>
<tr>
<td>STRAIN2</td>
<td>3.2</td>
<td>17</td>
<td>2.1</td>
</tr>
</tbody>
</table>

IT - Threshold value
FIG. 2.3a Original Image of Top Surface of Piston Head

FIG. 2.3b Thresholded Image of Fig. 2.3a Using Edge Operator ROB
   (CPU Time: 2.0 sec)
background are highly correlated. As can be seen in Figure 2.4a, the interior of the object is not uniform. This gives rise to false threshold values. Further work using this technique led us to the conclusion that the technique does not yield proper threshold values when there is no good separation between the object and the background.
FIG. 2.3c Thresholded Image of Fig. 2.3a Using Edge Operator DIF

(CPU Time: 4.2 sec)

FIG. 2.3d Thresholded Image of Fig. 2.3a Using Edge Operator LAP

(CPU Time: 3.1 sec)
FIG. 2.4a Original Image of the Surface of a Piston Head

FIG. 2.4b Thresholded Image of Fig. 2.4a Using Scatter-Plot Technique
2.3 **HISTOGRAM MAPPING TECHNIQUE**

2.3.1 **Introduction**

Hall [13] and Woods [14] have proposed that a transformation of the histogram of an image, using a transformation function equal to the cumulative distribution of its pixels, could have a considerable enhancement effect on the contrast of the image. This approach could also be used in interactive image enhancement by directly specifying a histogram and mapping it onto an image, in order to yield an image whose histogram has the specified distribution. This was tried by Janjua [15]. The technique is considered as previous work done in this area.

2.3.2 **Gray Level Transformation**

In order to process a given image such that the histogram of the resulting image has the shape of the (previously specified, desirable histogram, the first step is to equalize the gray levels of the original image.

2.3.2.1 **Gray Level Equalization [13],[16]**

Let \( r \) and \( s \) be the normalized gray levels in the original and equalized image respectively.

\[
0 \leq r \leq 1 ; \quad 0 \leq s \leq 1
\]  \hspace{1cm} (2.4)
Let \( p_r(r) \) and \( p_s(s) \) be the probability density functions of the original and transformed images respectively. It has been shown in [13] and [16] that it is possible to transform the histogram of an image to one having a uniform distribution using the following function:

\[
s = T(r) = \int_0^r p_r(w) \, dw \quad 0 \leq r \leq 1
\]  

(2.5)

This transformation function produces an image where \( p_s(s) \) is uniform in the interval \( 0 \leq s \leq 1 \). It has also been shown that this result does not depend upon the density function in the integrand of Equation 2.5. This implies that if the transformation function described in Equation 2.5 is performed on any distribution—ideally speaking, the result obtained would be the same, i.e., a uniform density within the given interval. This observation is brought to use in transforming an image such that its histogram acquires a desirable shape. In order to process images on a digital computer, the concepts described earlier must be formulated in discrete form as follows:

\[
p_r(r_k) = \frac{n_k}{n} \quad 0 < r_k < 1
\]

\[
k = 0, 1, \ldots, L-1
\]

(2.6)

where \( L = \) total number of levels.
\[ s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \quad 0 < r_k < 1 \]

\[ k = 0, 1, \ldots, L-1 \quad (2.7) \]

It is seen that gray level equalization of an image results in an improved contrast, since the dynamic range of the pixels in the image has been increased.

2.3.3 Direct Specification of a Bimodal Histogram

It has been demonstrated [10] that it is possible to transform an image to one whose histogram has any desirable distribution. A method of transforming the histogram of an image (where the task of selecting a good threshold would be quite difficult) to a shape where threshold selection would not pose a problem is illustrated.

A bimodal histogram like the one shown in Figure 2.5 is specified. Such a histogram is constructed using two Gaussian distributions having different means and standard deviations. Threshold selection for an image whose histogram is of the shape of Figure 2.5 is quite straightforward. The threshold is selected at the bottom of the valley between the two peaks, since this level separates the image into two sub-populations, i.e., the 'background' and the 'objects'. Our aim therefore is to
obtain an image whose histogram has the shape of Figure 2.5. This is made possible by equalizing the actual and specified histograms. The image is transformed by mapping the two histograms. The transformation process can be performed as shown in the following sections.
FIG. 2.5 A Specified Bi-Modal Histogram

FIG. 2.6a Original Image of Surface of a Piston Head
2.3.4 **Histogram Mapping** [16]

Let $p_x(r)$ and $p_z(z)$ be the original and desired probability density functions, respectively. The histogram of the original image is equalized using the transformation function $T(r)$ as given by Equation 2.7. The desired histogram is also equalized using another transformation, say, $G(z)$ as shown in Equation 2.8.

$$
V = G(Z) = \int_{0}^{Z} p_z(w) \, dw \quad (2.8)
$$

It is however known that the probability densities $p_s(s)$ and $p_v(v)$ would be identical uniform densities since the final result of histogram equalization is independent of the density inside the integral. Thus if an inverse transformation $z = G^{-1}(s)$, where $s$ is the uniform levels obtained from original image, the resulting values of $z$ would have the desired probability density function.

2.3.5 **Results of Single-level Thresholding by Histogram Mapping**

Figure 2.6a shows the original image of the surface of a piston head. The histogram of this image appears as shown in Figure 2.6b. It is obvious from the figure that the histogram appears more to be like a unimodal distribution. The image was mapped to a specified histogram consisting of two unimodal Gaussian distributions shown Figure 2.5. The
resultant transformed image obtained appears as shown in Figure 2.6c. Now by thresholding this image at a pre-determined threshold value a thresholded image, as shown in Figure 2.6d, is obtained. It can be seen that the image has been properly thresholded. However, this technique is very much sensitive to the specified histogram. Figure 2.7a shows another bimodal specified histogram with different mean and standard deviation values from the one shown in Figure 2.5. The corresponding transformed image obtained is shown in Figure 2.7b. When this image is thresholded at the pre-determined threshold value, it appears as shown in Figure 2.7c. It is seen that the threshold is far from correct. This drawback was felt in many other cases that were tested. Also, the technique is highly time-consuming, since mapping has to be performed from one histogram to another for the entire image. Again, the mapping has to undergo a search algorithm to find the proper transformed pixel values.
FIG. 2.6b Histogram of Image Shown in Fig. 2.6a

FIG. 2.6c Transformed Image of Fig. 2.6a Using Specified Histogram Shown in Fig. 2.5
FIG. 2.6d Thresholded Image of Fig. 2.6c

FIG. 2.7a A Specified Bi-Modal Histogram With Different Mean and Standard Deviation
FIG. 2.7b Transformed Image of Fig. 2.6a Using Specified Histogram in Fig. 2.7a

FIG. 2.7c Thresholded Image of Fig. 2.7b
2.3.6. **Multi-Level Thresholding**

It is found in certain cases that using binary thresholding for an image would result in a considerable loss of information. For such images it is necessary that instead of transforming the image into two distinct gray levels, we use more than two levels to separate the 'background' from the 'object'.

While dealing with images having multiple populations, it may be impossible to select one good threshold level for the entire image. Therefore in cases like these it is prudent to select more than one threshold level. Assigning various levels for each region would yield an image where the objects would be separated from the background.

Practically, when considering the entire histogram, we may not encounter a clearly multi-modal histogram and it may be difficult to locate the peaks and valleys in the histogram. Therefore, a section of the entire histogram is considered at a time, and a bimodality is searched for. If we select a threshold, then another section of the histogram from that threshold is investigated for bimodality. In this way, the entire image is thresholded at various levels.
2.3.7 **Direct Specification of Multi-Modal Histogram**

As in the case of directly specifying a bimodal histogram for single-level thresholding of an image, consider the specification of the histogram shown in Figure 2.8a.

This histogram consists of three Gaussian distributions with different means but same standard deviation. If we can obtain an image whose histogram has the distribution shown in Figure 2.8a, we could easily select two thresholds. As in the case of a bimodal histogram, here also the image can be transformed into one whose histogram has the distribution shown in Figure 2.8a. This would overcome the difficulty encountered while thresholding the original image at more than one level. Figure 2.8c shows the effect of this transformation on the original image shown in Figure 2.8b.

The result of multi-level thresholding of the transformed image is shown in Figure 2.8d.
FIG. 2.8a A Specified Multi-Modal Histogram

FIG. 2.8b Original Image of Surface of a Piston Head
FIG. 2.8c Transformed Image of Fig. 2.8b Using Specified Histogram in Fig. 2.8a

FIG. 2.8d Thresholded Image of Fig. 2.8c
2.3.8 Results of Multi-Level Thresholding Using Histogram Mapping

The specified histogram can be constructed by combining two or more unimodal Gaussian distributions. The histogram is constructed using the Equation 2.9.

\[
p(x) = \sum_{k=1}^{NT+1} \frac{P_k}{\sqrt{2\pi}\sigma_k} \exp \left( -\frac{(x - \mu_k)^2}{2\sigma_k^2} \right)
\]

(2.9)

where NT is number of thresholds required.

Each unimodal Gaussian distribution is governed by three parameters, viz, the mean (\(\mu\)), standard deviation (\(\sigma\)) and the a priori probabilities of the levels (\(P\)). The selection of these values mostly determine the transformed image and hence the thresholded image. The effect of variations in these values on the thresholded image is shown in Figures 2.9b, 2.9d, 2.9f for the multi-level thresholded images using specified histograms in Figures 2.9a, 2.9c, and 2.9e. It is clear that a deviation in any of the selected parameters could have a drastic effect on the resultant thresholded image.
FIG. 2.9a A Specified Multi-Modal Histogram

FIG. 2.9b Thresholded Image Obtained Using Specified Histogram in Fig. 2.9a
FIG. 2.9c A Specified Multi-Modal Histogram

FIG. 2.9d Thresholded Image Obtained Using Specified Histogram in Fig. 2.9c
FIG. 2.9e A Specified Multi-Modal Histogram

FIG. 2.9f Thresholded Image Obtained Using Specified Histogram in Fig. 2.9e
2.4 VARIABLE THRESHOLDING

2.4.1 Introduction

Chow and Kaneko [17] have proposed a method of variable thresholding for image segmentation. In this method, the image is divided into smaller windows; a gray level histogram is computed for each window; and thresholds are selected for those windows that have bimodal histograms. These thresholds are then interpolated to derive a variable threshold for the entire image. This method is particularly useful in cases where there is a large variation in the gray scale from one part of the image to another, so that a single fixed threshold cannot be used for the entire image. Chow and Kaneko successfully applied this method to detect the heart region on chest X-rays. Makaqawa and Rosenfeld [18] applied this technique on TV images of machine parts. The technique, as applied by Makaqawa et al. is explained in the following section.

2.4.2 Variable Thresholding Based on Local Bimodality

The first step in this method is to divide the image into smaller windows. This is as shown Figure 2.10a where the original image is of size 128 by 128 and the windows are of size 32 by 32.

The method used to determine bimodality and to select thresholds for the bimodal histograms was based on a process
<table>
<thead>
<tr>
<th>1,1</th>
<th>1,2</th>
<th>1,3</th>
<th>1,4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>2,2</td>
<td>2,3</td>
<td>2,4</td>
</tr>
<tr>
<td>3,1</td>
<td>3,2</td>
<td>3,3</td>
<td>3,4</td>
</tr>
<tr>
<td>4,1</td>
<td>4,2</td>
<td>4,3</td>
<td>4,4</td>
</tr>
</tbody>
</table>

\[ \text{FIG. 2.10a Segmentation of Image for Variable Thresholding} \]

\[ \text{FIG. 2.10b Threshold Computation for Each Pixel} \]
of Gaussian fitting. This method consists of the following steps that are carried out on each window of the image.

Step (a):

The mean and standard deviation of the histogram are computed. These are defined by

\[ \mu = \frac{1}{N} \sum F(i) \times i \]  \hspace{1cm} (2.10)

\[ \sigma = \sqrt{\frac{1}{N} \sum F(i) \times (i - \mu)^2} \]  \hspace{1cm} (2.11)

where \( F(i) \) is the histogram value for the gray level \( 'i' \), and \( N \) is the number of points in the window (32*32). The summation is carried out over the range of gray levels (in this case 0 to 255). If \( \sigma \ll 3 \) no threshold is computed for the histogram. If not, then a threshold computation is carried out as follows.

Step (b):

A least squares fit \( \phi_1 \)

---

3 The value indicated is obtained by testing on several images.
\[ f(i) = \frac{p_1}{\sigma_1} \exp \left[ -\frac{(i - \mu_1)^2}{2 \sigma_1^2} \right] + \frac{p_2}{\sigma_2} \exp \left[ -\frac{(i - \mu_2)^2}{2 \sigma_2^2} \right] \]  

(2.12)

to the histogram \( F(i) \) is found by adjusting the parameters \( p_1, \mu_1, \sigma_1, p_2, \mu_2, \) and \( \sigma_2 \). This is done as follows:

1. The histogram is smoothed by taking a local weighted average:

\[ F(i) = \frac{F(i-2) + 2F(i-1) + 3F(i) + 2F(i+1) + F(i+2)}{9} \]  

(2.13)

On the smoothed histogram, the deepest valley \( IV \) is found, and is used to divide the histogram into two parts. Using this the values of the parameters are calculated as follows.

2.

\[ N_1 = \sum_{i=0}^{IV} F(i) \quad N_2 = \sum_{i = IV+1}^{31} F(i) \]  

(2.14)

\[ \mu_1 = \frac{1}{N_1} \sum_{i=0}^{IV} F(i) \cdot i \quad \mu_2 = \frac{1}{N_2} \sum_{i = IV+1}^{31} F(i) \cdot i \]  

(2.15)
\[ \sigma_1 = \sqrt{\frac{1}{N_1} \sum_{i=0}^{IV} F(i) \ast (i - \mu_1)^2} \]  \hspace{1cm} (2.16)

\[ \sigma_2 = \sqrt{\frac{1}{N_2} \sum_{i=IV+1}^{3L} F(i) \ast (i - \mu_2)^2} \]  \hspace{1cm} (2.17)

\[ P_1 = \frac{N_1 \sigma_1}{\sum_{i=0}^{IV} \exp \left[ -\frac{(i - \mu_1)^2}{2 \sigma_1^2} \right]} \]  \hspace{1cm} (2.18)

\[ P_2 = \frac{\sum_{i=IV+1}^{3L} \exp \left[ -\frac{(i - \mu_2)^2}{2 \sigma_2^2} \right]}{2.19} \]

\[ \mu_2 - \mu_1 > 4 \]  \hspace{1cm} (2.20)

\[ 0.1 < \frac{\sigma_1}{\sigma_2} < 0.1 \]  \hspace{1cm} (2.21)

\[ \delta_{12} < 0.8 \]  \hspace{1cm} (2.22)

where \( \delta_{12} \) the valley-to-peak ratio is defined by
\[ \delta_{12} = \frac{\text{Min of } f \text{ in } [\mu_1, \mu_2]}{\text{Min } [f(\mu_1), f(\mu_2)]} \]

(2.23)

If these tests are not satisfied, no threshold is selected for that window. If not, a threshold is selected using the Gaussian fitting \( f(i) \). The threshold \( T \) is computed from the Equation 2.24.

\[
\left( \frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} \right) t^2 + 2 \left( \frac{\mu_2}{\sigma_2} - \frac{\mu_1}{\sigma_1} \right) t + \frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2} + 2 \ln \left( \frac{P_2 \sigma_1}{P_1 \sigma_2} \right) = 0
\]

(2.24)

After thresholds have been selected using the above procedure for bimodal windows, the thresholds are defined for the other windows by a local weighted averaging process. Let \( T(u,v) \) be the threshold assigned to the window centered at \((u,v)\). Then for a window centered at \((x,y)\), for which no threshold is assigned, a threshold value is computed from its neighbouring windows as follows:
\[
(T(X+1,Y) + T(X-1,Y) + T(X,Y+1) + T(X,Y-1))
\]

\[
(1/2) * (T(X+1,Y+1) + T(X+1,Y-1) + T(X-1,Y+1) + T(X-1,Y-1))
\]

At least one of the non-diagonal neighbouring windows must have a threshold assigned to it.

Finally, a threshold to each individual point is computed by bilinear interpolation of the window thresholds. If \( P \) is the point, surrounded by four windows centered at A, B, C and D as shown Figure 2.10b then the threshold for \( P \) is taken as

\[
\frac{1}{(a+b)(c+d)} \quad (bd \cdot T_A + bc \cdot T_B + da \cdot T_C + ca \cdot T_D) \quad (2.25)
\]

If \( P \) is not surrounded by four windows then the threshold of the nearest window is used.

2.4.3 Results of Variable Thresholding Scheme

The technique was applied as explained in the previous section. Figure 2.11a shows the original image of the surface of a piston head. The resultant thresholded image using Variable thresholding scheme is shown in Figure 2.11b. It can be seen that the image has been properly thresholded. Another example is given in Figure 2.12a and 2.12b showing a
proper segmentation. However the technique fails when the images are of low contrast. Figure 2.13a shows the image of a piston head. When thresholded using the Variable thresholding technique the image appears as shown in Figure 2.13b. We can see that the segmentation is not correct. This has occurred because the original image is of very low contrast. Further this technique, due to its excessive computation, needs a lot of execution time to obtain a thresholded image.
FIG. 2.11a Original Image of Surface of Piston Head

FIG. 2.11b Thresholded Image of Fig. 2.11a Using Variable Thresholding
FIG. 2.12a Original Image of Three Wooden Blocks

FIG. 2.12b Thresholded Image of Fig. 2.12a Using Variable Thresholding
FIG. 2.13a Original Image of Surface of a Piston Head

FIG. 2.13b Thresholded Image of Fig. 2.13a Using Variable Thresholding
2.5 IMAGE THRESHOLDING USING GAUSSIAN FITTING TECHNIQUE

2.5.1 Single-level Thresholding Using Gaussian Fitting Technique

As seen in the previous technique, the Variable thresholding scheme works, in its true sense, only in the interior of the image. For pixels on the border of the image, the technique uses the threshold of the nearest window. Also we see that the technique requires a considerable amount of computation time. Hence it appears more appropriate to apply the Gaussian fitting procedure on a global level i.e. for the entire image. However in this case a few modifications are required to make the technique more general. The range of gray levels are extended from 0-31 to 0-255. Again the condition for bimodality is not included and the threshold is computed for Gaussian fitting. Also the averaging process of the histogram is modified such that a false valley may not be detected. For a group of IG gray levels, the levels are replaced by their average value. In this manner it possible to eliminate zero values and hence avoid the detection of a false valley.

2.5.2 Results of Gaussian Fitting Technique

The Gaussian fitting technique was applied to several images and it yielded satisfactory results in most of the cases. Figure 2.14a shows the original image of a piston
head. The thresholded image obtained using the Gaussian fitting technique is shown in Figure 2.14b. Another example is shown in Figure 2.15a, which is the image of a strain gauge. The thresholded image appears as shown in Figure 2.15b. However, the technique fails in some of the cases. Figure 2.16a shows the original image of another strain gauge. The thresholded image however eliminates most of the object as shown in Figure 2.16b. The reason for such a result is that the image contains multiple populations. This is clear from the histogram of the image shown in Figure 2.16c. It shows that the image consists of at least three populations, and hence requires at least two thresholds for a proper segmentation. Hence the technique is extended for multi-level thresholding as explained in the following section.
FIG. 2.14a Original Image of Surface of a Piston Head

FIG. 2.14b Thresholded Image of Fig. 2.14a Using Gaussian Fitting Technique
FIG. 2.15a Original Image of a Strain Gauge

FIG. 2.15b Thresholded Image of Fig. 2.15a Using Gaussian Fitting Technique
FIG. 2.16a Original Image of Another Strain Gauge

FIG. 2.16b Thresholded Image of Fig. 2.16a Using Gaussian Fitting Technique
2.5.3 Multi-level Thresholding Using Gaussian Fitting

The original histogram of the image is averaged out again as in the case of single-level thresholding. Let NT be the number of thresholds required by the image. Then using the averaged histogram, NT + 1 peaks are located in the histogram. Then taking two peaks at a time, the NT deepest valleys are obtained. Using these results, the multi-modal Gaussian fitting is performed as follows:

Step (a):

Compute the parameters of the Gaussian fitting, $N_K$, $P_K$, $\mu_K$, and $\sigma_K$ as given below.

\[
N_K = \sum_{i=IV_{K-1}}^{IV_K} F(i) \quad (2.26)
\]

\[
\mu_K = \frac{1}{N_K} \sum_{i=IV_{K-1}}^{IV_K} F(i) \cdot i \quad (2.27)
\]

\[
\sigma_K = \sqrt{\frac{1}{N_K} \sum_{i=IV_{K-1}}^{IV_K} F(i) \cdot (i - \mu_K)^2} \quad (2.28)
\]

\[
P_K = \frac{N_K \sigma_K}{\exp \left[ - \frac{(i - \mu_K)^2}{2 \sigma_K^2} \right]} \quad (2.29)
\]

$K = 1, \ldots, NT + 1$

and NT is Number of Thresholds.
Using these values now a Gaussian fitting can be performed by using Equation 2.30.

\[ p(x) = \frac{p_K}{\sigma_K} \exp \left[ -\frac{(x - \mu_K)^2}{2 \sigma_K^2} \right] \]  \hspace{1cm} (2.30)

\[ K = 1, \ldots, NT + 1 \]

Now by taking two unimodal distributions at a time the threshold between them is computed using Equation 2.31.

\[ \left( \frac{1}{\sigma_K^2} - \frac{1}{\sigma_{K+1}^2} \right) T_K^2 + 2 \left( \frac{\mu_{K+1}}{\sigma_{K+1}} - \frac{\mu_K}{\sigma_K} \right) T_K \]

\[ + \frac{\mu_K^2}{\sigma_K^2} - \frac{\mu_{K+1}^2}{\sigma_{K+1}^2} + 2 \ln \left( \frac{p_{K+1} \sigma_K}{p_K \sigma_{K+1}} \right) = 0 \]  \hspace{1cm} (2.31)

\[ K = 1, \ldots, NT + 1 \]
Using these threshold values the image can then be segmented into multiple regions.

2.5.4 Results of Multi-Modal Gaussian Fitting Technique

The original image of the strain gauge used in the single-level thresholding is again used in this case. The Gaussian fitting for the histogram is obtained as shown in Figure 2.16d. It is seen that the fitting follows the histogram very closely identifying all the three regions. The multi-level thresholded image obtained using this fitting is shown in Figure 2.16e. It is easily seen that the thresholded image has selected more features than the single-level thresholding. Thus the multi-modal Gaussian fitting technique proves to be useful in cases where the single-level threshold does not yield proper results.
FIG. 2.16c Histogram of Image Shown in Fig. 2.16a

FIG. 2.16d Multi-Modal Gaussian Fitting For Image Shown in Fig. 2.16a
FIG. 2.16e Multi-Level Thresholded Image of Fig. 2.16a Using Gaussian Fitting Technique
2.6 AN ITERATIVE THRESHOLDING SCHEME

2.6.1 Introduction

Object-background discrimination can be improved by deriving a threshold from a series of background samples taken close enough to the object to exclude most background clutter but not close enough to include the object. Ideally the background close to the object can then be used to find the mean background level and the object region can be used in obtaining the mean object gray level. Given these two values the discrimination can then be readily achieved. The process can be repeated to improve these two values which is used in deriving the threshold for the image. Widler and Calvard [19] proposed this technique of thresholding in an attempt to threshold images of handwritten text. The technique was applied to the TV images of machine parts as discussed in section 2.6.3.
FIG. 2.17a Initial Estimation of Thresholded Image

FIG. 2.17b Schematic Diagram for the Iterative Processor
2.6.2 Iterative Threshold Selection

Suppose an object is located within a square image of picture elements. Without any previous knowledge about the image, it can be assumed that the four corners of the square contain the background while the rest can be considered as object. Such an image is shown in Figure 2.17a. This patch can then be used as a switching function \( f(s) \) to route the digitized image into one of the two integrators (object and background). The mechanism can be described with reference to the block diagram shown in Figure 2.17b. The signal controlling the switch is referred to as the switching function \( f(s) \), which in fact is the thresholded image. If \( f(s) = 0 \), the corresponding pixel of the input image is sent to the background integrator. If \( f(s) = 1 \) the pixel is sent to the object integrator. When all the pixels of the image have been considered in this manner, the integrator outputs are averaged to obtain the mean object and background gray levels. A threshold can now be computed for the image by taking the average of these two mean gray levels. The thresholded image, obtained using this value, now becomes our new switching function.

The operation is repeated until the mean gray levels of the object and background remain a constant, in other words the threshold remains a constant. It is to be noted that the technique does not involve any histogram calculations.
2.6.3 Results of Iterative Thresholding Scheme

Figure 2.18a shows the original image of the surface of a piston head. Figure 2.18b shows the thresholded image obtained using the iterative thresholding scheme. It can be seen that the threshold value computed through this technique is close to the correct threshold. As another example Figure 2.19a shows the original image of a strain-gauge. The thresholded image obtained by the iterative scheme appears as shown in Figure 2.19b. However, the technique fails to yield proper results when applied on images with more than two distinct populations. For example Figure 2.20a shows the original image of the surface of another piston head. The histogram of this image appears as shown in Figure 2.20b. It is easily seen that the image contains more than two distinct populations. Hence the thresholded image using the iterative technique appears as shown in Figure 2.20c. Another example where the technique fails is shown in Figure 2.21a and 2.21b. This is the image of a transmission gear wheel.

The reason for this failure is that, in the attempt to segment the image strictly into two regions, the third population is forced into either one of the groups thereby giving rise to erroneous results. Hence it is necessary to segment the image into more than two groups in order to obtain meaningful results. As a result this technique of iterative thresholding has been extended to multi-level
FIG. 2.18a Original Image of Surface of Piston Head

FIG. 2.18b Thresholded Image of Fig. 2.18a Using Iterative Technique
FIG. 2.19a Original Image of Strain Gauge

FIG. 2.19b Thresholded Image of Fig. 2.19a Using Iterative Technique
FIG. 2.20a Original Image of Surface of a Piston Head

FIG. 2.20b Histogram of the Image Shown in Fig. 2.20a
FIG. 2.20e Thresholded Image of Fig. 2.20a Using Iterative Technique
FIG. 2.21a Original Image of a Section of a Gear Wheel

FIG. 2.21b Thresholded Image of Fig. 2.21a Using Iterative Technique
thresholding. The procedure is explained in the following section.

2.6.4 **Iterative Multi-Level Threshold Selection**

In this procedure, when the details of the image are not known, an initial estimation of the multi-level thresholded image can be made as shown in Figure 2.22a. However in cases such as surface flaw detection of similar objects it is possible to use the multi-level thresholded image obtained for one part, as an initial estimation for all other parts.

The schematic diagram for the processor is as shown in Figure 2.22b. The operation can be explained as follows. The initial estimation shown in Figure 2.22a is used as the switching function $f(s)$ shown in Figure 2.22b. Consider the pixel $p(i,j)$ in the original image. If the corresponding pixel in $f(s)$ is, say, zero then the pixel $p(i,j)$ is sent to integrator I. If the value of $f(s)=1$ then $p(i,j)$ is sent to integrator II and when $f(s)=2$, $p(i,j)$ is sent to integrator III. The process is repeated for all the pixels in the original image. Now by averaging the integrators I, II, III it is possible to obtain the mean gray levels of three different populations present in the image. Taking two subsequent mean values together and finding their average yields the two different threshold values $T1$ and $T2$. These values are now used to threshold the original image. The
thresholded image thus obtained becomes the new switching function. The procedure continues until both the threshold values remain a constant.

The above procedure is explained for a two level thresholding. However the same procedure can be generalized for 'N' number of thresholds by using 'N+1' number of integrators. The iterative process continues until all the threshold values remain constant simultaneously.
FIG. 2.22a Initial Estimation of a Multi-Level Thresholded Image

FIG.2.22b Schematic Diagram of Iterative Technique for Multi-Level Thresholding
2.6.5 **Results of Multi-level Thresholding by Iterative Scheme**

The same cases considered in single-level thresholding can be used to test for multi-level thresholding. Figure 2.23 shows the multi-level thresholded image obtained by the iterative technique for the original image shown in Figure 2.20a. Figure 2.24 shows the multi-level thresholded image of Figure 2.21a. We see that the segmentation of the image is proper in these cases than the single-level thresholding.
FIG. 2.23 Multi-Level Thresholded Image of Fig. 2.20a Using Modified Iterative Technique

FIG. 2.24 Multi-Level Thresholded Image of Fig. 2.21a Using Modified Iterative Technique
2.7 CONCLUSIONS ON THRESHOLDING

2.7.1 Comparison of Various Thresholding Schemes

In our attempt to select an optimum thresholding technique, we have looked into several techniques that use different approaches. It is however necessary to select only the optimum technique that yields proper threshold values in most of the cases. Also since the technique is to be applied to manufactured parts normally to be inspected for surface defects in a short time (<3 secs) it is necessary that the technique yields the proper threshold value with the least amount of computational time.

As seen before, the histogram mapping procedure proposed by Janjua due to its inherent nature of mapping each pixel, requires excessive computational time. Also the results of the technique are very sensitive to the statistical values used in the specified histogram. Hence it cannot be considered for our purpose. The results of all other techniques have been grouped together in Table 2.3 along with the proper threshold values. Also for cases where multi-level thresholding is possible, they have been added in the list for comparison. It is easily seen that the Gaussian fitting technique yields the optimum results with least computational time.
### TABLE 2.3

Comparison of Various Thresholding Schemes

<table>
<thead>
<tr>
<th>Images</th>
<th>Iterative</th>
<th>Gaussian</th>
<th>Scatter</th>
<th>Variable</th>
<th>Correct</th>
<th>Threshold</th>
<th>Fitting</th>
<th>Plot</th>
<th>Thresholding</th>
<th>Value</th>
<th>Technique Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>PISTON</td>
<td>S - 65</td>
<td>S - 58</td>
<td>S - 76</td>
<td>FALSE</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEAD</td>
<td>4.7 SEC</td>
<td>1.3 SEC</td>
<td>2.0 SEC</td>
<td>5.3 SEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEAR</td>
<td>M - 46,92</td>
<td>S - 36</td>
<td>S - 80</td>
<td>FALSE</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WHEEL</td>
<td>12.3 SEC</td>
<td>1.2 SEC</td>
<td>2.0 SEC</td>
<td>5.5 SEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PISTON</td>
<td>M - 43,117</td>
<td>S - 37</td>
<td>S - 103</td>
<td>TRUE</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEAD</td>
<td>18.0 SEC</td>
<td>0.8 SEC</td>
<td>2.0 SEC</td>
<td>5.8 SEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRAIN</td>
<td>M - 30,63</td>
<td>M - 26,62,71</td>
<td>S - 42</td>
<td>FALSE</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAUGE</td>
<td>11.5 SEC</td>
<td>2.1 SEC</td>
<td>2.0 SEC</td>
<td>5.7 SEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S - Single-level Thresholding
M - Multi-level Thresholding
TRUE - Thresholded image acceptable
FALSE - Thresholded image not acceptable
2.7.2 **Conclusions**

From the results shown above it can be concluded that the Gaussian fitting technique is the optimum technique for our purpose of thresholding images. However, by using proper initial estimations the iterative technique can also be used for this purpose. It can be seen that the results of the two techniques are very close. However, the iterative technique needs more computational time than the Gaussian fitting technique. All our further testing was carried out using the thresholded images obtained by the Gaussian fitting technique.
Chapter III

BORDER FOLLOWING

3.1 INTRODUCTION

Thresholding techniques were investigated in the previous chapter. The thresholded image by itself can contribute very little for the purpose of image analysis. In most cases it is desirable to obtain the boundaries of the various regions present in an image. Such a task can be accomplished in several ways. By convolving the image with proper edge templates it is possible to obtain the boundaries between the various regions of the image. It is also possible to locate the boundaries by scanning the entire image, looking for gray level transitions. A brief description of these procedures is given below.

3.1.1 Edge Detection Using Template Matching

Consider a 3 by 3 image region as shown in Figure 3.1. We can define $G_x$ as

$$G_x = (g + 2h + i) - (a + 2b + c) \quad (3.1)$$

and $G_y$ as

$$G_y = (c + 2f + i) - (a + 2d + g) \quad (3.2)$$
The gradient at the point \( e \) is then defined as
\[
G = \left( G_x^2 + G_y^2 \right)^{\frac{1}{2}}
\] (3.3)

By comparing the Equation 3.1 and Figure 3.1 it can be seen that \( G_x \) is the difference between the first and third row of the 3 by 3 image with the elements closer to \( e \) weighted twice as much as the corner values. Thus it can be said that the function \( G_y \) represents an estimate of the derivative in the \( x \) direction. Similarly \( G_x \) can be considered as an estimate of the derivative in the \( y \) direction. The implementation of this gradient approach is made simpler by means of template matching. Consider the two templates shown in Figure 3.2a and 3.2b. It can be immediately seen that the two templates when convolved would yield the values of \( G_x \) and \( G_y \). Thus by using these two templates it is possible to detect any horizontal and vertical edges present in an image. In other words it can be used to detect the boundaries between the regions in an image.

Such an approach was tried on the thresholded image shown in Figure 3.3a and the result obtained is shown in Figure 3.3b. It can be seen that the technique results in broken edges. Also the edges obtained are too thick, that a thinning algorithm has to be run to get the proper boundaries.
FIG. 3.1 A Sample 3 by 3 Image Region For Sobel's Gradient

\[
\begin{array}{ccc}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{array}
\]

FIG. 3.2a A Horizontal Edge-Detection Template

\[
\begin{array}{ccc}
  -1 & -2 & -1 \\
  0 & 0 & 0 \\
  1 & 2 & 1 \\
\end{array}
\]

FIG. 3.2b A Vertical Edge-Detection Template

\[
\begin{array}{ccc}
  -1 & 0 & 1 \\
  -2 & 0 & 2 \\
  -1 & 0 & 1 \\
\end{array}
\]
FIG. 3.3a Thresholded Image of a Piston Head

FIG. 3.3b Edge-Detection of Fig. 3.3a Using Sobel's Gradient.
3.1.2 **Edge Detection by Scanning**

The boundaries between regions in a thresholded image can also be obtained by scanning the image. For example, consider the thresholded image as shown in Figure 3.4a. The image is scanned from top to bottom and from left to right. Whenever a difference in gray level is observed a boundary point is placed at that location. In this manner the whole boundary of the regions in an image can be obtained as shown in Figure 3.4b.

The approach was tried on a thresholded image shown in Figure 3.5a and the resulting boundary followed image is shown in Figure 3.5b. It can be seen that the boundaries have been clearly traced out. Also the boundaries are not thick as in the case of template matching. However by this approach it is not possible to isolate boundaries of different regions. This information is important for calculations of area, perimeter, centroid etc of the different regions. Also the information can be used for such applications as; shape analysis to identify the shape of the region isolated by the boundary. Hence it becomes necessary to find some method by which to isolate the coordinates of the boundaries of different regions. Such an algorithm is described in Ref [4] which can follow boundaries in single-level thresholded images and also obtain the co-ordinates of those boundaries. The modified
### FIG. 3.4a A Sample Thresholded Image

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### FIG. 3.4b Edge-Detection in Fig. 3.4a By Scanning

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
algorithm developed by Chottera and Shridhar [20], which can also detect and eliminate spurious features, that are often generated due to surface reflections, is employed in this work.
FIG. 3.5a Thresholded Image of a Gear Wheel

FIG. 3.5b Edge-Detected Image of Fig. 3.5a Scanning
3.2 **BOUNDARY FOLLOWING IN SINGLE-LEVEL THRESHOLDED IMAGES**

In this procedure the image is thresholded using a threshold value obtained by the Gaussian fitting technique. The two levels are identified as 0 and 1. The border following can then be applied as explained in Ref [20].

The algorithm has been developed in such a manner that it can detect transitions either from 0-1 or from 1-0. However the algorithm interprets in such a way that both the transitions appear to be from a 1-0.

In this algorithm, as it can be seen, the border point once detected results in the tracing of the entire border of the region. Hence the border points of any region can be easily stored in an array to be used later. Also, the method of computing the neighbouring point co-ordinates makes the algorithm computationally efficient in detecting borders. The algorithm is tested out on several single-level thresholded images. As an example, consider the thresholded image of a piston head shown in Figure 3.6a. The border following algorithm yields the borders of various regions as shown in Figure 3.6b. It is seen that the borders are properly identified. However, the original algorithm was developed to work only on single-level thresholded images. To detect borders of regions in multi-level thresholded images it necessary to modify slightly the algorithm as explained in the following section.
TABLE 3.1

Co-ordinates of eight neighbours.

<table>
<thead>
<tr>
<th></th>
<th>LX (J)</th>
<th>LY (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>JD</td>
</tr>
<tr>
<td>2</td>
<td>LX (1) + K1</td>
<td>LY (1) - K2</td>
</tr>
<tr>
<td>3</td>
<td>LX (2) - K2</td>
<td>LY (2) - K1</td>
</tr>
<tr>
<td>4</td>
<td>LX (3) - K2</td>
<td>LY (3) - K1</td>
</tr>
<tr>
<td>5</td>
<td>LX (4) - K1</td>
<td>LY (4) + K2</td>
</tr>
<tr>
<td>6</td>
<td>LX (5) - K1</td>
<td>LY (5) + K1</td>
</tr>
<tr>
<td>7</td>
<td>LX (6) + K2</td>
<td>LY (6) + K1</td>
</tr>
<tr>
<td>8</td>
<td>LX (7) + K2</td>
<td>LY (7) + K1</td>
</tr>
</tbody>
</table>

Co-ordinates of the border element (I1, J1)
Co-ordinates of the first neighbour (ID, JD),

\[ K1 = JD - J1 \]
\[ K2 = ID - I1 \]
FIG. 3.6a Thresholded Image of a Piston Head

FIG. 3.6b Border Followed Image of Fig. 3.6a
3.3 **BOUNDARY FOLLOWING IN MULTI-LEVEL THRESHOLDED IMAGES**

Consider an image thresholded at two levels. The three different gray levels in the image can be marked as 0, 1 and 2. For following the borders in this image the steps can be outlined as follows:

1. Detect the first border element at \((I_1, J_1)\). In this case the transition can occur in any manner. The immediately neighbouring element \((ID, JD)\) is determined as in the single-level thresholded case.

2. Starting with \((ID, JD)\) and proceeding clockwise, label the other seven neighbours of \((I_1, J_1)\) as 2, 3,..., 8. Set \(k = 2\).

3. Store the first border element value in, say, **MARK1**.

4. Evaluate the co-ordinates \(LX(K), LY(K)\) of the \(k\) neighbour of \((I_1, J_1)\) using Table 3.1

5. If the pixel at the \(k\)-th neighbour is equal to **MARK1**, then this pixel is the next border element. Define this as \((I_1, J_1)\) and the immediately preceding neighbour as \((ID, JD)\) and go to Step 2.

6. If the pixel at the \(k\)-th neighbour is not equal to **MARK1** then set \(k = k + 1\) and go to Step 4.

7. Proceed until the first border element detected in Step 1 is encountered again.

The above algorithm was tried on several multi-level thresholded images. As an example consider the multi-level
thresholded image of the gear wheel as shown in Figure 3.5a. The border following algorithm, when applied on this image yields the image shown in Figure 3.7. It can be seen clearly that the borders of all regions have been properly identified. Again in this case it is possible to obtain the co-ordinates of each region independently. This data can then be used for calculation of, say, each contact area of the gear wheel. As another example consider the multi-level thresholded image shown in Figure 3.8a. The border followed image appears as shown in Figure 3.8b indicating a clear isolation of boundaries.

The above procedure can be easily extended to more than two levels of gray.
FIG. 3.7 Border-Followed Image of Fig. 3.5a
FIG. 3.8a  Multi-Level Thresholded Image of a Piston Head

FIG. 3.8b  Border-Followed Image of Fig. 3.8a
Chapter IV

TEMPLATE MATCHING

4.1 INTRODUCTION

In Digital Image Processing, the term 'Template' is defined as an array designed to detect some invariant regional property [16].

As an introduction to template matching let us consider a 3 by 3 window (or mask) with its weights represented by \( w_1, w_2, \ldots, w_9 \) as shown in Figure 4.1. Let \( x_1, x_2, \ldots, x_9 \) be the gray levels of the pixels inside the mask. Template matching is then defined as the inner product of these two vectors

\[
W = \begin{bmatrix}
  w_1 \\
  w_2 \\
  \vdots \\
  w_9 
\end{bmatrix}
\]

and

\[
X = \begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_9 
\end{bmatrix}
\]

(4.1)
and the inner product is

\[ w'x = w_1 x_1 + w_2 x_2 + \ldots + w_9 x_9 \]

A feature is said to be identified when this inner product exceeds a specified threshold value \( T \) i.e.

\[ w \cdot x > T \]

In general the template or mask can be of any \( N \) by \( M \) dimension depending on the problem requirements.

As explained in Section 1.5 the most straightforward procedure is a point detection. The next level of complexity involves detection of lines in an image. Figure 4.2a shows the template to detect a vertical line of one pixel thickness. The response would be strong when it is passed over a line with a constant background. Similarly Figure 4.2b, c and d show the templates that can be used to detect lines oriented in three different directions.

The template matching procedure can also be used to detect edges present in an image as seen in the previous chapter. Frei and Chen [21] have proposed a technique where a simultaneous application of all the templates can be done, and the type of region can be analysed from the response.
FIG. 4.1 A Point Template

FIG. 4.2 Vertical Line Template

FIG. 4.2b Horizontal Line Template

FIG. 4.2c Angled Line Template (-45°)

FIG. 4.2d Angled Line Template (+45°)
4.2 **TEMPLATE MATCHING AS APPLIED IN THE SYSTEM**

As discussed in the introduction, after thresholding and border following it is necessary to perform template matching in order to locate inherent features in the image. In the problems considered (piston head, gear wheel and strain gauge) the inherent features to be detected contain a straight edge at their boundaries. Hence a line template is used to detect these edges and thereby locate the positions of the inherent features in the image.

4.3 **DIRECT TEMPLATE MATCHING**

Consider the sample border followed image shown in Figure 4.3a, and a line template, shown in Figure 4.3b. The maximum response is obtained when the template exactly coincides with the line in the sample image. By using a threshold value \( T = 2^N \), where \( N \) is the length of the template, it is possible to locate the straight line by thresholding the response. Template matching can be performed efficiently using either the Fourier transform or the Number Theoretic Transform as explained in the following section.
FIG. 4.3a A Sample Border-Followed Image

<table>
<thead>
<tr>
<th>-1</th>
<th>2</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
</tbody>
</table>

FIG. 4.3b A Vertical Line Template
4.4 Template Matching Using Fast Fourier Transform

Let $f(x)$ be a continuous function of a real variable $x$. The Fourier Transform of the function is defined as

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp \left[ - j 2\pi ux \right] dx \quad (4.2)$$

Similarly for a function $f(x,y)$ of two variables, that is continuous and integrable, the Fourier transform is defined as

$$f(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \exp \left[ - j 2\pi (ux + vy) \right] dx \, dy \quad (4.3)$$

However, since the signals that are dealt with are discrete in nature, it is necessary to have the above relations in discrete form.

Let $f(x)$ be a discrete function of length $N$. The Discrete Fourier transform of this function is defined as

$$F_n(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp \left[ - j 2\pi \frac{ux}{N} \right] \quad (4.4)$$

Similarly for a two dimensional function $f(x,y)$ of size $M$ by $N$ the Discrete Fourier transform is defined by the Equation

$$F(u,v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \exp \left[ - j 2\pi \frac{ux}{N} \right] \exp \left[ - j 2\pi \frac{vy}{N} \right] \quad (4.5)$$
In the case of digital images normally the value of \( M \) and \( N \) are the same.

The Fourier transform can be applied to the convolution or correlation two signals. The convolution of two functions \( f(x) \) and \( g(x) \), both one dimensional and continuous, is defined as

\[
\begin{align*}
\quad \quad f(x) \ast g(x) &= \int_{-\infty}^{\infty} f(\alpha) \, g(x - \alpha) \, d\alpha \\
\quad \quad & \quad \quad \quad \quad \text{(4.6)}
\end{align*}
\]

However the convolution theorem states that convolution in time domain is transformed into multiplication in frequency domain and vice versa. In other words, if \( f(x) \) and \( g(x) \) are two functions of a real variable \( x \) and \( F(u) \) and \( G(u) \) are their Fourier transforms respectively. Then the convolution theorem states that

\[
\begin{align*}
\quad \quad f(x) \ast g(x) & \iff F(u) \cdot G(u) \\
\quad \quad F(u) \ast G(u) & \iff f(x) \cdot g(x)
\end{align*}
\]

where \( \ast \) represents a convolution operation and \( \cdot \) represents a multiplication operation. This shows us that the complex operation of convolution is transformed into a simple operation of multiplication by means of using the Fourier transform. The Fourier transform of discrete signals and its inverse transform are both periodic in nature. Hence the product of two transforms is also a periodic function. This makes it necessary to select the
length of the period such that there is no wraparound effect. If \( f(x) \) is a discrete function of length \( M \) and \( g(x) \) is another discrete function of length \( N \) then it is necessary to select the length of the period of convolution as \( L \), where

\[
L > M + N - 1
\]

Based on this, we can now define the discrete convolution of \( f(x) \) and \( g(x) \) as

\[
f(x) * g(x) = \sum_{m=0}^{M-1} f(m) g(x-m) \tag{4.7}
\]

Similarly the relation for convolution of two dimensional discrete signals is defined as

\[
f(x, y) * g(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) g(x-m, y-n) \tag{4.8}
\]

The relations for correlation of one and two dimensional signals is very much similar except for a change in sign. The equation for two dimensional discrete correlation is given as

\[
f(x, y) * g(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) g(x+m, y+n) \tag{4.9}
\]
From the above relations it is clear that the correlation operation is the same as that of template matching. If \( f(x, y) \) is the two dimensional signal of an image, and \( g(x, y) \) is the template then by computing their Fourier transforms \( F(u, v) \) and \( G(u, v) \) and performing an inverse transform on the product of \( F(u, v) \) and \( G(u, v) \) we obtain a template matched image.

Consider an image of size 128 by 128 and a template of size 7 by 3 (in our case a line template). Then it is necessary to select a period of 134 by 130 as the convolution period. The same operation can also be performed by means of sectioned convolutions. In this case the entire image is sectioned into smaller regions and then convolved with the template. The outputs can then be added together or attached to each other depending on the type of method (overlap save and overlap add). The operation is however computationally expensive due to the complex arithmetic. Hence it becomes necessary to investigate other possible ways of convolving the image with the template.

4.5 TEMPLATE MATCHING USING NUMBER THEORETIC TRANSFORM

As shown in the above section, the Fast Fourier Transform method of template matching results in operations in the complex domain. This is computationally expensive and also has extensive memory requirements for large amount of data. In the Fast Fourier Transform the kernel of the
transformation is the \( N \) roots of unity in the complex domain, where \( N \) is the transform length.

In residue number theory it is possible to obtain the \( N \)th root of unity by means of cyclic group generators \([22]\). Thus it is possible to perform the same convolution operation by using this kernel without going into the complex domain. The procedure for convolution in the Number Theoretic Transform can be explained as follows:

1. Obtain the NTT of the image using \( \mathbb{Z}_n \) moduli where \( n \) varies from 2 to \( N \), \( N \) being the number of moduli used. It is necessary that the product of the \( N \) moduli is sufficiently large enough to prevent overflow in the operation.
2. Repeat the above operation, this time with the template instead of the image.
3. Obtain the product of the NTT of the image and the template for each modulus.
4. Multiply the result with the multiplicative inverse or the transform length twice for each modulus.
5. Perform an inverse NTT on the above result.
6. Convert the result back to decimal form using the residues by any of the conversion procedures available.

The above procedure is computationally inexpensive since this does not involve any complex domain operations.

However, the procedure still requires extensive memory since
the NTT of the image and template with respect to various moduli have to be stored. Also when the number of moduli increases the computational time increases. Hence it would be convenient to have some form of hardware that could perform this entire operation. A hardware NTT convolver that was designed and built by Naqpal [23] at the University of Windsor can be used for this purpose. The procedure is explained as follows.

4.5.1 **Template Matching Using Hardware NTT Convolver**

The hardware NTT convolver has been built in such a way that it can be used for filtering the image with any set of filter coefficients. In the case of template matching the filter coefficients are replaced by the template to be convolved with the image. However, the following steps are essential to perform this operation.

1. Compute the NTT of the filter coefficients (or template) to be convolved.
2. Transfer the NTT of filter coefficients (or template) to the convolver.
3. Transfer the image to be convolved to the convolver.
4. Start the filtering operation by sending the proper control signals.
5. Read back the filtered image from the convolver.

The convolver can be also used for larger templates by using the overlap-save technique of convolution. In this
case the image is sectioned according to the requirements. These sections are then transferred one after another to the convolver and the results are properly stored.

4.5.2 Results of Template Matching Using NTT Convolver

The border followed image of the surface of a piston head appears as shown in Figure 4.4a. This image was convolved with a vertical line template shown in Figure 4.3b using the hardware NTT convolver. The resultant image appears as shown in Figure 4.4b. It can be seen that the vertical line of the eyebrows have been clearly identified by the the template. As another example consider the border followed image shown in Figure 4.5a. This image is that of a section of a gear wheel discussed previously. This image again was convolved with a vertical line template shown in Figure 4.3b. The resultant image is given in Figure 4.5b. It can be seen that the vertical edges of the teeth have been clearly isolated from the rest of the image. As an example of a different template the following case was considered. A combination of horizontal and vertical templates was taken for this purpose. The template appears as shown in Figure 4.6a. This template was again convolved with the border followed image shown in Figure 4.5a. The result of the convolution is given in Figure 4.6b. It can be seen that both horizontal and vertical sections of the image have been isolated.
As discussed earlier, the template matching procedure can be used in locating some of the inherent features in the image. From the above results it can be seen that this is possible by using the hardware NTT convolver. The speed of operation of template matching using NTT convolver is only .5 secs for an image of size 128 by 128. Thus inherent features can be easily located at a much faster rate by means of the hardware convolver.
FIG. 4.4a Border Followed Image of a Piston Head

FIG. 4.4b Template Matched Image of Fig. 4.4a
FIG. 4.5a Border Followed Image of a Gear Wheel

FIG. 4.5b Template Matched Image of Fig. 4.5a
FIG. 4.6a Horizontal and Vertical Line Template

COLOURED PICTURES
Images en-couleur

FIG. 4.6b Template-Matched Image of Fig.4.5a Using Template Shown in Fig.4.6a
Chapter V
SYSTEM INTEGRATION

5.1 INTRODUCTION

As mentioned in the beginning of this thesis, it is required to develop a system that can detect surface flaws by means of Digital image processing. Having investigated the various algorithms that are required for the system it is now necessary to integrate the algorithms to form a complete system.

5.2 TOUCH SCREEN

As shown earlier in the block diagram of Figure 1.1 it is necessary to have some form of switch that indicates the type of part that is currently sent for inspection. In an actual industrial environment this selection is normally performed by a shop-floor worker, and hence it is required to have a simple and easy means of identification. The touch screen facilitates the above task. The various parts that can be tested by the algorithm is displayed on the screen of the display terminal. The touch screen is mounted on the face of this terminal. The employee can be made to select the appropriate part by merely touching that part on the screen.
A touch screen developed by the TSD* data products has been used for this purpose. This consists of the actual screen and the interface unit that connects the screen to the host computer. Whenever the screen is touched, the position of the touch is sensed as voltage difference and the information is sent to the interface. The interface in turn transforms this data into the corresponding x and y co-ordinates of the position and sends it as ASCII data to the host computer. An algorithm has been developed, that displays the menu for selection and waits for the data from the interface. Once the x and y co-ordinates are obtained the information is used in identifying the part for testing.

As an example Figure 5.1 shows the menu pad displaying the three parts that can be tested by the algorithm. By touching one of them the algorithm does the rest of the job in testing the part.

* The Touch Screen was manufactured by:
TSD DISPLAY PRODUCTS, INC
302, Legget Drive,
Kanata, Ontario K2K 1V5,
Canada.
FIG. 5.1 A Sample Menupad
5.3 DISPLAY UNIT

As mentioned above it is necessary to have a display terminal in the system. It is useful in displaying the menu pad for selection of the part for testing. Again it can be used later in displaying the results. A color graphics terminal developed by AYDIN CONTROLS\textsuperscript{5} has been used for this purpose. Algorithms have been developed to display the images on this terminal. Also an algorithm for displaying the menu pad for the touch screen has been developed and used in the testing of parts. The terminal can also be used for displaying the histograms of images which is useful in developing the algorithms.

5.4 CONVOLVER

As mentioned in Chapter IV it is necessary to have a convolver for fast template matching purposes. Also it is sometimes necessary to filter the images in the preprocessing stage before it can be analysed. Hence, it is very useful to have a hardware convolver that can suit both purposes. This is made possible by a hardware NTT convolver designed and built by Haqpal.\textsuperscript{6}

\textsuperscript{5} The color graphics terminal was manufactured by:
AYDIN CONTROLS
414, Commerce Drive,
Fort Washington,
PA 19034,
USA
Algorithms have been developed for transferring data to and from the convolver. Algorithms have been developed to filter images of size 128 by 128. For larger size images such as 256 by 256 the sectioned convolution using the overlap-save technique has been adopted. Algorithms have also been developed for template matching using the convolver. Here again an algorithm for templates of size up to 60 by 60 using overlap save technique has been developed.

The results of the template matching using the convolver has already been discussed in Section 4.5.2.

The information obtained by using the convolver can now be used in locating inherent features in the image. This is explained in sections 5.5.2 and 5.5.3.

5.5 **PROBLEM TESTING**

5.5.1 **Introduction**

Having looked at the complete system we shall now see how problems could be tested by this system. A flow chart explaining the flow of the procedure is shown in Figure 5.2. The information obtained from the touch screen is used to identify the part for testing. Once identified, then each individual part is tested in a different way. The procedure is explained below.

---

6 Dr. Nagpal designed and built the NTT convolver used in this work. Dr. Nagpal obtained his Ph D from the University of Windsor in June 1981 under the supervision of Dr. G. A. Jullien.
FIG. 5.2 Flow of the Algorithm

1. Obtain digitized input image
2. Threshold the image
3. Follow the borders in the image

III. Type of part?
   I. Compute number of regions in image
   II. Locate and remove inherent features in images
   III. Locate the reference point in image

- Compute area and number of regions in the image
- Compute area and location of contact area

Classify part and display results
5.5.2 Testing of Piston Head

The original image of the part to be inspected, in this case a piston head, is obtained by means of a vidicon camera. In real time processing this image can then be stored in the memory of the host computer. However for the case of testing the image is stored in cartridge disk. The program for parts inspection can then later read this image from the disk file. The image thus obtained is then thresholded by means of the Gaussian fitting technique. The thresholded image so obtained is then used in the border following technique to obtain the boundaries of various regions in the image and also their areas. Now there are two different approaches that can be used to locate the inherent features in the image. The boundary following algorithm can also record the number of pixels in a boundary. This information can be used to identify the inherent features, viz the eyebrows in the piston head. The other approach is to identify the straight line portions of the eyebrows by means of template matching with a vertical line template. The template matching response is maximum only at the location of the straight line portion of the image which can then be used to identify the eyebrows.

7 Real time here means the rate at which the part is required by the assembly line.
Figure 4.4a shows the original image of the surface of a piston head and the corresponding template matched image is shown Figure 4.4b. It is seen that the straight line portion of the eyebrows have been clearly identified. However this procedure can be avoided in this case since the identification can also be easily done by the number of pixels on the boundary of the eyebrow. Hence the algorithm has been developed using this information (number of pixels on the boundary) to identify the eyebrows.

Figure 5.3a shows the result of the algorithm. The image on the top left corner shows the surface of a piston head. The image that is adjacent to it, on the right, shows the corresponding thresholded image. The final image that is obtained after removing the eyebrows is shown on the bottom left corner. It is seen that the faults in the image have been identified. The results for this part is shown in Figure 5.3b. The time required for this inspection, is only approximately 3 secs on the 32 bit machine SEL computer. However the algorithm requires more computational time when other algorithms are used for thresholding purposes. As another example, a different piston head is shown in Figure 5.4a and b.

From the above results it can concluded that it is feasible to detect surface flaws by means of digital image processing.
FIG. 5.3a Displayed Results for a Piston Head

PART INSPECTED: PISTON HEAD

THE # OF REGIONS ARE ... 37

THE TOTAL AREA OF FAULTS IS ... 216,000,000

HRS: 13  MIN: 14  SEC: 14  THRESHOLD START
HRS: 13  MIN: 14  SEC: 15  THRESHOLD END
HRS: 13  MIN: 14  SEC: 17  ANALYSIS START
HRS: 13  MIN: 14  SEC: 19  ANALYSIS END

FIG. 5.3b A Sample Report of the Results for Piston Head
FIG. 5.4a Results of Another Piston Head.

PART INSPECTED: PISTON HEAD

THE # OF REGIONS ARE ... 4

THE TOTAL AREA OF FAULTS IS ... 11,000,000

HRS: 13  MIN: 16  SEC: 44  THRESHOLD START
HRS: 13  MIN: 16  SEC: 45  THRESHOLD END
HRS: 13  MIN: 16  SEC: 48  ANALYSIS START
HRS: 13  MIN: 16  SEC: 49  ANALYSIS END

FIG. 5.4b Report of the Results Shown in Fig. 5.4a
5.5.3 Testing of Gear Wheel

The picture of a gear wheel is shown in Figure 5.5a. The digitized image of a section of a gear wheel appears as shown in Figure 5.5b. As seen in the image there are some dark patches on the teeth of the gear wheel. These patches represent the contact area of the gear wheel when it is allowed to run against another gear wheel. It is important that the area of contact lies in the centre of the tooth for efficient and long life of the wheel. Hence it is required to find this area of contact and its location with respect to some reference point. Based on this information the quality of the wheel can then be analysed. By image processing techniques it is possible to locate such flaws. The algorithm for this part is as follows.

The original image of the gear wheel is obtained through the vidicon camera. This image is then thresholded to separate the dark patches from the rest of the image. The thresholded image is then border followed to obtain the boundaries of these contact areas. Now it is necessary to obtain a reference point to judge on the location of the contact area. For this purpose it is necessary to use some inherent feature of the image. It can be seen from Figure 5.5b that the edges of the teeth appear to be vertical in nature. Also these edges come out as straight lines when it is border followed. Hence by applying a vertical template
FIG. 5.5a  A Differential Gear Wheel

FIG. 5.5b  Digitized Image of a Section of a Gear Wheel
it is possible to locate this straight portions in the image. The image obtained by such a template matching is shown in Figure 4.5b. As discussed above the straight lines are clearly isolated from the rest of the image. Thus using the starting pixel of this vertical edge as reference point it is possible to judge on the location of the contact area also the border following algorithm computes the area of the contact region. Hence by using the area and location of the contact area it is possible to judge on the quality of the part. Figure 5.6a shows the combination of all the above operations by the algorithm. The results regarding the area and location is computed and is given in Figure 5.6b.
FIG. 5.6a Displayed Results for the Gear Wheel

PART INSPECTED: TRANSMISSION GEAR WHEEL

THE AREA OF CONTACT IS ... 404.500000000

THE CENTROID OF CONTACT AREA IS ... (41, 70)

DISPLACEMENT IN X ...

DISPLACEMENT IN Y ... 35

HRS: 13   MIN: 17   SEC: 47   THRESHOLD START
HRS: 13   MIN: 17   SEC: 49   THRESHOLD END
HRS: 13   MIN: 17   SEC: 51   ANALYSIS START
HRS: 13   MIN: 17   SEC: 53   ANALYSIS END

FIG. 5.6b A Sample Report of the Results of Gear Wheel
5.5.4 **Testing of a Strain Gauge**

The original image of a strain gauge appears as shown in Figure 5.7. In this case it is required to find any broken connections in the circuit of the strain gauge chip. Again image processing techniques can be used for this purpose. The algorithm followed is given below.

The original image of the strain gauge is passed through a thresholding scheme. This thresholding procedure yields the connectors clearly isolated from its background. In cases where the single level thresholding fails it is possible to use multi-level thresholding. The thresholded image thus obtained is then passed through a border following algorithm. By this the boundaries of the different sections of the connectors are available. The number of connectors in the strain gauge chip is a fixed value for a given batch of chips. Hence this information can be given as input to the algorithm. The border following algorithm yields us also the number of regions present in the image. This can now be compared against the number of regions given as input. An increase in the number of regions would then indicate that there is a broken connector in the chip. The testing of this algorithm is shown in Figure 5.8a. The original image appears on the left top corner of the image. The corresponding thresholded image is shown in the right top corner. The border followed
image is shown in the bottom of the image. The results obtained from the algorithm is listed as shown in Figure 5.8b. As it is shown the number of regions has been clearly computed. Based on this information the quality of a strain gauge can be determined. Figure 5.9a and 5.9b show the results of testing on another two strain gauges. It is once again clear that the connector have been clearly isolated and counted.
FIG. 5.7 Image of a Strain Gauge
FIG. 5.8a Displayed Results for a Strain Gauge

PART INSPECTED: STRAIN GAUGE

THE # OF REGIONS IS ... 7

HRS: 13 MIN: 18 SEC: 47 THRESHOLD START
HRS: 13 MIN: 18 SEC: 49 THRESHOLD END
HRS: 13 MIN: 18 SEC: 51 ANALYSIS START
HRS: 13 MIN: 18 SEC: 53 ANALYSIS END

FIG. 5.8b A Sample Report of the Results of a Strain Gauge
FIG. 5.9a Displayed Results for Another Strain Gauge

PART INSPECTED: STRAIN GAUGE

THE # OF REGIONS IS ... 6

HRS: 13  MIN: 19  SEC: 47  THRESHOLD START
HRS: 13  MIN: 19  SEC: 50  THRESHOLD END
HRS: 13  MIN: 19  SEC: 52  ANALYSIS START
HRS: 13  MIN: 19  SEC: 54  ANALYSIS END

FIG. 5.9b Report of Results Shown in Fig.5.9a
Chapter VI
SUMMARY AND CONCLUSIONS

6.1 SUMMARY OF THE WORK DONE

The problem considered in this work is: "Surface Flaw Detection Using Image Processing Techniques". An outline is developed to solve this problem by image processing (as shown in Figure 5.2). The major problem of the algorithm is to obtain a proper thresholded image. It is important that the automatic thresholding scheme yields an accurate result since the rest of the algorithms are based on the thresholded image. Several methods for automatic thresholding are carefully investigated, and tested on actual problems. The constraints placed on these techniques are accuracy of results and computational time. It is found that the Gaussian fitting technique proposed Chow and Kaneko [17] applied on a global level yielded the best results with the least amount of computational time. All the algorithms were initially tested on the NOVA 840 mini-computer and later tried on the dedicated machine viz SEL 32/27. The results appeared as shown in Table 2.3 In some of the cases a single level thresholded image did not yield proper segmentation. For such cases the algorithm is extended to multi-level thresholding.
The thresholded images thus obtained were then used by a border following algorithm. For cases of multi-level thresholded images the border following algorithm is extended for multi-level thresholded images.

In some of the cases taken for testing it is required to apply a template matching algorithm to locate some inherent features in the image. For this purpose algorithms are developed to perform template matching using a high speed hardware NTT convolver. The algorithm has been developed to perform template matching with templates of sizes up to 60 by 60.

Also as mentioned previously it is necessary to preprocess the image in some of the cases before other image processing algorithms can be applied. Hence the hardware NTT convolver developed by Nagpal is used for this purpose. Algorithms are developed for computing NTT of filter coefficients. Also algorithms are developed to transfer data to and from the convolver. The algorithms are developed in such a manner that images of sizes up to 256 by 256 can be convolved with a given set of filter coefficients.

In the process of developing the hardware for NTT convoler, peripheral units such as PROM programmers are required to be included in the system. Algorithms are written for data transfer between the PROM programmer and the host computer.
As shown in the initial block diagram for a complete system it is required to have display systems and hard copy units. For this reason algorithms are developed to display images on a graphics terminal, AYDIN, and also for making hard copy units by means of plotting the images on a TRILOG color printer/plotter.

An easy method of identifying the part for inspection has been made possible by means of a touch screen. Algorithms are developed to scan and analyse the data from the touch screen, and use it in identifying the part for inspection.

Finally, all these individual algorithms are grouped together for complete surface flaw analysis.

6.2 CONCLUSIONS

From the above work it is concluded that Image Processing Techniques can be applied for surface flaw detection. By this approach it is possible to have an automated system for inspection of surface defects. The set-up eliminates possible manual errors that can occur in the normal manual inspection.

---

* The TRILOG color printer/plotter was manufactured by:
  TRILOG, INC
  17391, Murphy Ave,
  Irvine, CA 92714,
  USA
REFERENCES


OTHER REFERENCES


VITA AUCTORIS

1958 Born on 2nd of May in Coimbatore, India.

1974 Completed Matriculation at Carmel Garden Matriculation School, Coimbatore, India.

1975 Completed Pre-University Course at PSG Arts College, Coimbatore, India.

1980 Completed Bachelor of Engineering (Electronics and Communication) at PSG College of Technology, Coimbatore, India.

1980-81 Studied Master of Engineering (Computer Science) at PSG College of Technology, Coimbatore, India.

1983 Candidate for the Degree of Master of Applied Science in Electrical Engineering at the University of Windsor, Canada.