Design of vision tracking system.

Jun. Yang

University of Windsor

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

Recommended Citation
https://scholar.uwindsor.ca/etd/1669

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.
Design of Vision Tracking System

By

Jun Yang

A Thesis
Submitted to the Faculty of Graduate Studies and Research
Through the Department of Electrical and Computer Engineering in Partial Fulfillment
Of the Requirements for the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada
2002

©2002 Jun Yang
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-80517-4
© 2002 Jun Yang

All Rights Reserved. No part of this book may be reproduced, in any form or by any means, without the prior permission in writing from the author.
Abstract

In this thesis, hardware and software design of a vision tracking system is presented. The whole system is a low-cost, flexible 2D device. Three algorithms are implemented in this system for position deviation detection and stretching behavior analysis.

The first algorithm, which is based on the statistic approaching method, is developed in this thesis. The other two algorithms, which are based on cross correlation method and the directional flow-change (DFC) method, respectively, are existing algorithms. In this research, both algorithms are modified, implemented and tested.

The experiments and the results presented in this thesis show that the first algorithm is a simple, fast, robust and accurate method to estimate the position and the size of the object. The cross correlation based algorithm is suitable for pattern recognition. But for position deviation detection, it is not reliable and need to be improved. The third algorithm is efficient for position deviation detection, and it works well for certain type of shapes.

This system can be used as a platform for developing, testing and improving new algorithms.
Acknowledgments

I would like to thank my supervisor Dr. Xiang Chen for his guidance, comments, reviews and support during the period of my study and research.

I also wish to thank Dr. Chunhong Chen and Dr. Arunita Jaekel for their suggestions and comments on my research work and for their willing to serve as the committee members. My thanks also go to Dr. Erfani for chairing my defense.

Some of my fellow graduate students were particularly helpful throughout this research. Among them I wish to name Ms. Hongmei Gao for the help I have received from both her excellent previous work and herself. My thanks are delivered to Mr. Alan Johns for providing me with equipments and to the Technology Support Centre for helping building the platform. My special thanks go to Brahm Industry in Windsor for their technical support at the early stage and, the last but not least, to Manufacturing and Material Ontario (MMO) for their financial support for the whole program.

Finally, I would like to thank my wife and my daughter for help, encouragement and understanding of my working habits. My wife also gave me a lot of suggestions on my software development.
# Table of Contents

Abstract ........................................................................................................... iv

Acknowledgements ......................................................................................... v

List of Figures ................................................................................................. viii

List of Tables ................................................................................................ X

Chapter 1  Introduction .................................................................................... 1

1.1 Introduction ............................................................................................. 1
1.2 Objective of My Thesis ........................................................................... 3
1.3 Thesis Organization ................................................................................. 5

Chapter 2  Preliminary Theory ......................................................................... 6

2.1 Introduction ............................................................................................. 6
2.2 Statistical Approach .............................................................................. 6
2.3 Cross Correlation Analysis ................................................................... 8
2.4 Directional Flow-Change Method ......................................................... 9
  2.4.1 Chain Coding .................................................................................. 9
  2.4.2 Directional Flow-Change ............................................................... 10
  2.4.3 Shape Description ......................................................................... 12
2.5 Proportional Controller ......................................................................... 13

Chapter 3  Algorithm Design and Control Strategy ..................................... 14

3.1 Introduction ............................................................................................. 14
3.2 Image Preprocessing .............................................................................. 15
  3.2.1 3 x 3 Neighborhood Smoothing .................................................. 15
3.2.2 3×3 Median Filter ........................................................................15
3.2.3 Adaptive Threshold Method for Image Enhancement ...............16

3.3 Segmentation ..................................................................................16
3.3.1 The Existing Segmentation Method ...........................................17
3.3.2 New Segmentation Method .........................................................19

3.4 Feature Extraction and Image Analysis .........................................21
3.4.1 Statistical Approach .................................................................22
3.4.2 Method Based on Cross Correlation .........................................24
3.4.3 Directional Flow Change Method ..............................................25

3.5 Review of The Three Algorithms .................................................30
3.5.1 The Flowchart of the Statistical Approach Method ..................30
3.5.2 The Flowchart of the Cross Correlation Method .......................31
3.5.3 The Flowchart of the Directional Flow-Change Method ............31

3.6 Control Strategy .............................................................................32

Chapter 4 Hardware Implementation .................................................34
4.1 Introduction ....................................................................................34
4.2 Lighting .........................................................................................34
4.3 Lens ..............................................................................................36
4.4 CCD Camera ................................................................................37
4.5 DT3120 Image Grabber .................................................................49
4.6 Motorola MC68HC912B32 Microcontroller ..................................44
4.7 Tower Hobbies TS-69 S3K Servo Motor ......................................46
4.8 Rail Rack for Tracking ..................................................................48
4.9 Hardware Implementation of The System .....................................49

Chapter 5 Software Implementation ................................................51
5.1 Introduction ....................................................................................51
5.2 DT3120 SDK ...............................................................................51
5.3 Microsoft Vision SDK ..................................................................52
5.4 Software Design ...........................................................................54
5.5 Embedded Software Design .........................................................62
Chapter 6 Experiments and Results .............................................. 65

6.1 Introduction ............................................................................. 65
6.2 Experiments with The Statistical Approach ....................... 65
6.3 Experiments with The Cross Correlation Based Algorithm .... 71
6.4 Experiments with The DFC Algorithm ................................. 72
6.5 Experiments with Tracking System ..................................... 81
6.6 Algorithm Analysis and Comparison ................................... 82
    6.6.1 Algorithm Based on Statistical Approach ..................... 82
    6.6.2 Algorithm Based on Cross Correlation ......................... 82
    6.6.3 Algorithm Based on DFC ........................................... 83
    6.6.4 Comparison of Three Algorithms ................................. 83

Chapter 7 Conclusion and Future Work ................................... 85

7.1 Conclusion ............................................................................. 85
7.2 Comments on Three Algorithms ....................................... 87
7.3 Future Work ........................................................................ 87

References .................................................................................... 89

Appendix A Picture of The System ........................................... 92
Appendix B VC++ Source Code (on floppy disk) ...................... 93
Appendix C C Code for MC68HC912B32 (on floppy disk) ......... 208
Appendix D Operation Manual .................................................. 215

Vita Auctoris ................................................................................. 222
List of Figures

Figure 1.1  Block Diagram of Assembly Line With Machine Vision System ............... 1
Figure 1.2  Block Diagram of A Basic Machine Vision System ............................. 2
Figure 1.3  Block Diagram of System Architecture ............................................ 4
Figure 2.1  Numbering Scheme for 8 Neighborhoods Chain Code .......................... 10
Figure 2.2  Block Diagram of Closed-Loop Control ............................................. 13
Figure 3.1  Block diagram of image processing steps .......................................... 14
Figure 3.2  $3 \times 3$ Neighborhood of Pixel $P_{ij}$ .................................................. 15
Figure 3.3  (a) The Image of The Foam Barrier  (b) The Interested Image Area ........ 16
Figure 3.4  Criteria of Existing Segmentation Method ....................................... 17
Figure 3.5  Special Case ......................................................................................... 19
Figure 3.6  1D Histogram segmentation method ..................................................... 19
Figure 3.7  The General 1D Histogram Function of Foam Barrier in x Direction ...... 20
Figure 3.8  Cropped Image of the Foam Barrier .................................................... 21
Figure 3.9  Special Cases for New Segmentation Method ..................................... 21
Figure 3.10 Statistical Approach Based Image Analysis ........................................ 22
Figure 3.11 Stretched Foam Barrier ....................................................................... 23
Figure 3.12 Cross Correlation Between Two Images .............................................. 25
Figure 3.13 (a) Cropped Image, (b) Enlarged Image .............................................. 25
Figure 3.14 Input Flow and Output Flow of The Pixel $P_i, J=4$ ............................ 26
Figure 3.15 $\delta$ Function of The Cropped Foam-Barrier Image .............................. 27
Figure 3.16 (a) Original Image, (b) Constructed polygon of The Foam Barrier ...... 28
Figure 3.17 Flowchart of The Statistical Approach Algorithm .............................. 30
Figure 3.18 Flowchart of The Algorithm Based on Cross Correlation ................. 31
Figure 3.19 Flowchart of The DFC Algorithm ....................................................... 32
Figure 3.20 Block Diagram of The Proportional Control ....................................... 33
Figure 4.1  Block Diagram of The Light Source.................................................35
Figure 4.2  Size of The Foam Barrier............................................................35
Figure 4.3  Imaging Principle of Thin Lens ....................................................36
Figure 4.4  Lens Distortion............................................................................37
Figure 4.5  Interlaced Transfer CCD Sensor..................................................37
Figure 4.6  Block Diagram of DT3120 Image Grabber .................................39
Figure 4.7  The Rear Panel of The DT3120 Image Grabber..........................40
Figure 4.8  Horizontal Video Signal...............................................................42
Figure 4.9  Vertical Video Signal.................................................................42
Figure 4.10 Spatial Relationship of Video Signal..............................................43
Figure 4.11 Monochrome Mode of DT3120....................................................44
Figure 4.12 MC68HC912B32 Single Chip Mode Memory Map.......................45
Figure 4.13 Serial Interface Block Diagram....................................................46
Figure 4.14 Control Signal and Modifying Servo Motors...............................47
Figure 4.15 Picture of The Rail Rack..............................................................48
Figure 4.16 The Block Diagram of The System..............................................49
Figure 4.17 Schematic of Servo Motor Control.............................................50
Figure 5.1  The Related Projects and The Classes in SDK.............................53
Figure 5.2  Structure of The Image RECT......................................................54
Figure 5.3  Block Diagram of Application Classes........................................54
Figure 5.4  Basic Windows Interface.............................................................55
Figure 5.5  Flowchart of The Main Routine....................................................56
Figure 5.6  RS232 Configuration Subroutine...................................................57
Figure 5.7  RS232 Configuration Dialog........................................................57
Figure 5.8  Grab Image Template Dialog......................................................58
Figure 5.9  Flowchart of The “Grab Template Dialog”....................................59
Figure 5.10 Flowchart of Timer subroutine....................................................60
Figure 5.11 Flowchart of Image Acquisition Subroutine................................61
Figure 5.12 Data Format of Communication between PC and Microcontroller...62
Figure 5.13 Flowchart of Embedded Control.................................................63
Figure 5.14 Flowchart of SCI Interrupt..........................................................65
Figure 5.15 Flowchart of TOC Interrupt.........................................................65
| Figure 6.1 | The Image of The White Square Paper ...........................................66 |
| Figure 6.2 | Effectiveness of Image Enhancement with Different Thresholds ..........66 |
| Figure 6.3 | The 1D Histograms $H(x), H(y)$ of The Square Image ......................66 |
| Figure 6.4 | Enhanced Image of The Bottle Cap, Threshold=150 ............................67 |
| Figure 6.5 | 1D Histograms of The Bottle Cap in x, y Direction ..........................67 |
| Figure 6.6 | Angles Between The Camera and The Cubic Object ...........................68 |
| Figure 6.7 | Relationship Between The Real Deviations and Estimated Deviations ...70 |
| Figure 6.8 | Input Images Before Chain Coding ..............................................72 |
| Figure 6.9 | (a) $\delta$ Function of The Image, (b) Constructed Polygon ...............72 |
| Figure 6.10 | The Image of The Rectangle with One Rough Edge ...........................73 |
| Figure 6.11 | (a) $\delta$ Function of The Image, (b) Constructed Polygon ...............73 |
| Figure 6.12 | The Image of The Foam Barrier ..................................................74 |
| Figure 6.13 | $\delta$ Function of The Foam Barrier ..........................................74 |
| Figure 6.14 | Constructed Polygon of The Foam Barrier ....................................75 |
| Figure 6.15 | $\delta$ Function of The Foam Barrier ..........................................76 |
| Figure 6.16 | Constructed Polygon ...............................................................76 |
| Figure 6.17 | The Image of The Bottle Cap .....................................................77 |
| Figure 6.18 | Three Calculated $\delta$ Functions of The Bottle Cap .......................77 |
| Figure 6.19 | Three Constructed Polygons of The Bottle Cap ..............................78 |
| Figure 6.20 | Relationship between Real Deviation and Estimated Deviation ..........80 |
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Angle Coding Scheme</td>
<td>12</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Modified Angle Coding scheme</td>
<td>29</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Comparison of The Position Deviation</td>
<td>69</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Comparison of Position Deviation in 1st Quadrant</td>
<td>71</td>
</tr>
<tr>
<td>Table 6.3</td>
<td>Results of Experiment 4</td>
<td>78</td>
</tr>
<tr>
<td>Table 6.4</td>
<td>Comparison of The Position Deviation</td>
<td>79</td>
</tr>
<tr>
<td>Table 6.5</td>
<td>Comparison of The Three Algorithms</td>
<td>81</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Introduction

As the result of significant progress in computer and integrated circuit (IC) technology, machine vision has been applied in many manufacturing processes since 1980’s. In some industries, such as semiconductor, electronics and automotive industry, many products can not be produced with high quality without machine vision. In Figure 1.1, an assembly line with machine vision system is illustrated.

![Figure 1.1 Block Diagram of An Assembly Line With Machine Vision System](image)

Machine vision is the use of optical devices for non-contact sensing, it automatically receives and interpret an image of a real scene in order to obtain information and/or
control actuators or processes. A basic machine vision system involves image interpretation, decision making and machine control.

![Block Diagram of A Basic Machine Vision System](image)

**Figure 1.2 Block Diagram of A Basic Machine Vision System**

As shown in Figure 1.1, a basic machine vision system includes the following unit: Light source, sensor, I/O interfaces, image processor and decision making and control unit. Generally, machine vision is an integration of image acquisition and processing technology. First, the sensor (usually, the camera) converts the optical signal into electrical signal, then, a computer or a dedicated IC is used for processing the image and making the decision [19]. The big challenge for researchers and developers is the computational power required to handle the large amount of image data with a required resolution and in a short time period. Algorithm and hardware architectures are two critical aspects in machine vision. A lot of algorithms for image processing and pattern recognition have been developed theoretically in recent years. In order to test the efficiency, accuracy, effectiveness and robustness of the algorithms, a real time platform is needed to test and verify the algorithms to explore the complexity and memory consumption of the algorithms.
According to the hardware architecture, machine vision systems can be sorted into three categories:

- VLSI dedicated machine vision system [4]
- Computer based machine vision system [1][3][17]
- DSP/FPGA based machine vision system [8][14][27][31]

In a VLSI dedicated machine vision system, the algorithm is implemented by hardware, so that the system can work faster than any other software systems. For a number of repeated applications, the cost can be reduced to an affordable level. If algorithm needs to be changed frequently, it is not flexible. Since Intel introduced Pentium MMX® CPU in 1997, more and more companies have adopted software solution instead of hardware solution. Because Pentium MMX® CPU provides industry with a powerful tools to realize complex algorithms with software at a satisfied speed. The computer based machine vision system is more flexible for applications to which special algorithms have to be applied [1][3]. The disadvantage is that the system needs more space to mount. DSP/FPGA based machine vision system has both advantages of VLSI dedicated system and computer based system, such as small size and flexibility. But, it is very difficult to employ DSP/FPGA based system for the algorithms that are more complex and need large amount of memory.

1.2 Objective of This Thesis

The motivation of this research comes from a real problem existing in the foam barrier assembly line for automobile door handle escutcheons. During the assembly process, a robotic arm is employed to catch the foam barrier on a conveyor. However, practical
difficulties are that the position of a foam barrier may deviate due to possible stretching behavior of foam material and the pre-configured robotic arm cannot handle this position uncertainty. Two algorithms based on 2-D gray level spatial image have been developed theoretically to conduct position analysis based on cross-correlation and directional flow change methods [5][6][7]. Regarding the research work presented in this thesis, there are three goals that need to be reached:

1. To built a platform for developing new algorithms.
2. To solve the position detection problem
3. To test, verify and improve the existing two algorithms [5][6][7].

To realize these three goals at low cost, we need a flexible system that can be employed either as a machine vision system that can potentially solve the real manufacturing problem or as a platform to develop and test new algorithm. As mentioned before, the computer based machine vision system is more flexible than any other system. The system architecture is illustrated in Figure 1.2.

![Figure 1.3 Block Diagram of System Architecture](image)
1.3 Thesis Organization

This thesis consists of seven chapters. The research background, motivation of the research and the current technology level in this area is provided in Chapter 1. Three basic methods for image processing and pattern recognition, involving statistical approach, cross correlation and the directional flow-change method are presented. The details of the three algorithms based on the methods introduced in Chapter 2 are described in Chapter 3. The hardware architecture and the features of the hardware components are introduced in Chapter 4. The software architecture and flowchart are covered in Chapter 5. The experiments and the results based on the presented algorithms are presented in Chapter 6. In Chapter 7, the tasks undertaken by this thesis is summarized and some recommendations for future work are given.
Chapter 2

Preliminary Theory

2.1 Introduction

In this chapter, three image-analysis methods in spatial domain are introduced for position deviation detection and stretching behavior analysis. First, the statistical approach method is described. The cross correlation analysis method and the DFC method are explained briefly. It is pointed out that the last two methods can identify the pattern of a planar shape. Last, a simple proportional control strategy is presented.

2.2 Statistical Approach

Statistical approach method is based on mean value and standard deviation that are important tools in statistics and are widely used in different areas. For image analysis, they are usually applied to estimating the image position and size [9][15][17].

For a given gray level image in spatial domain, two 1D histograms can be constructed along its x and y direction, respectively, which represent the spatial distributions of image-pixel intensities along the corresponding axes.

**Definition 2.1.1** Given a $M \times N$ image $f(x,y)$, the 1D histogram $H(x)$ in x direction is:

$$H(x) = \sum_y f(x, y); \quad (2.1)$$
the 1D histogram $H(y)$ in $y$ direction is:

$$H(y) = \sum_x f(x, y);$$  \hspace{1cm} (2.2)

**Definition 2.1.2** Given an image, where there is only one object, the center position $x_c, y_c$ can be defined as:

$$x_c = \frac{\sum_{x=0}^{W-1} xH(x)}{\sum_{x=0}^{W-1} H(x)};$$  \hspace{1cm} (2.3)

$$y_c = \frac{\sum_{y=0}^{H-1} yH(y)}{\sum_{y=0}^{H-1} H(y)};$$  \hspace{1cm} (2.4)

where, $W$ is the width of the image in pixels, $H$ is the Height of the image in pixels, $H(x)$ and $H(y)$ are the histogram distributions of the image along $x$ and $y$ directions, respectively.

**Definition 2.1.3** Given an image, where there is only one object, the size of the object $S_w$ and $S_H$ in pixels can be defined as:

$$S_w = f_x \sqrt{\frac{\sum_{x=0}^{W-1} (x-x_c)^2 H(x)}{\sum_{x=0}^{W-1} H(x)} };$$  \hspace{1cm} (2.5)

$$S_H = f_y \sqrt{\frac{\sum_{y=0}^{H-1} (y-y_c)^2 H(y)}{\sum_{y=0}^{H-1} H(y)} };$$  \hspace{1cm} (2.6)

where, $f_x$ and $f_y$ are constant scaling factors.
2.3 Cross Correlation Analysis

Cross-correlation analysis is one of the popular methods for template matching in machine vision application because of its accuracy and robustness. The maximum correlation value indicates the closest match [5][6][12][13].

**Definition 2.3.1** Given two \( M \times N \) matrices \( G \) and \( F \),

\[
G = \begin{bmatrix}
g_{11} & \cdots & g_{1N} \\
\vdots & \ddots & \vdots \\
g_{M1} & \cdots & g_{MN}
\end{bmatrix}, \quad F = \begin{bmatrix}
f_{11} & \cdots & f_{1N} \\
\vdots & \ddots & \vdots \\
f_{M1} & \cdots & f_{MN}
\end{bmatrix},
\]

the cross correlation of the matrices \( G \) and \( F \) is a \((2M-1)\times(2N-1)\) matrix \( R \),

\[
R = [r_{i,j}], \quad i = -M+1, \cdots, -1, 0, 1, 2, \cdots, M-1; \quad j = -N+1, \cdots, -1, 0, 1, 2, \cdots, N-1; \quad \text{and}
\]

\[
r_{i,j} = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f_{k,l} g_{k+i, l+j}, \quad g_{k-i, j-l} = 0 \quad \text{if} \begin{cases} k+i < 0 \text{ or } k+i \geq M \\ l+j < 0 \text{ or } l+j \geq M \end{cases}
\]  

(2.7)

If \( G \) and \( F \) represent two gray level images that are similar, the correlation analysis should result in the maximum value at the point of the best alignment. Therefore, the image \( G \) can be viewed as the same as image \( F \), if

\[
\max_{i,j} (r_{i,j}) \geq c \times \max(\text{autocorrelation});
\]

(2.8)

where, \( c \) is the threshold value.

In practical applications, the normalized cross correlation is employed in order to prevent false correlation peaks arising from the changes in the mean image gray level. The following equations are normalized cross correlation.

\[
k1 = \left[ \sum_{k=1}^{M} \sum_{l=1}^{N} f_{k,l}^2 \right]^{-1/2},
\]

(2.9)

\[
k2 = \left[ \sum_{k=1}^{M} \sum_{l=1}^{N} g_{k,l}^2 \right]^{-1/2},
\]

(2.10)
\[ C_g = \frac{r_g}{k2}; \]  
(2.11)

Because \( k_1 \) is a constant as long as the template does not change, it can be ignored in locating the maximum correlation.

Using the edged images to do correlation has the same effect as the normalized cross correlation method [5][6].

2.4 Directional Flow-Change Method (DFC)

The directional flow-change concept is employed for shape recognition. This method involves two steps. The first step is to construct the contour array \( \Phi \) based on chain coding method[5][7][10][18]. The second step is to get a set of side length and angle codes for image analysis by locating critical points, calculating side length, and coding angles based on the constructed contour. Before further introducing the DFC method, some concept has to be defined first.

The contour array \( \Phi \) is defined as:

\[ \Phi = \{ p_0, \ p_1, \ p_2, \ \cdots, \ p_{N-1} \}, \]

where \( N \) is the total number of contour points, \( p_0 \) and \( p_{N-1} \) are the starting point and the ending point, respectively.

\( \Phi_i(J) \) is defined as a portion of a contour consisting of \( 2J + 1 \) points \( p_{i-j}, \cdots, p_i, \cdots, p_{i+j} \)

where \( J = \gamma N \) with \( 0 < \gamma < 1 \). \( J \) is called the “supported length” and \( \gamma \) is called “supported rate” [7][10].

2.4.1 Chain Coding

Chain coding [18] is very popular coding technique, which can be used for describing the shape of the object. There are a number of chain coding algorithms for tracking the
region boundaries in a binary image. In this research, 8 neighborhood coding method is
used for tracking the contour of the object.

![Diagram of 8 Neighborhood Chain Code]

Figure 2.1 Numbering Scheme for 8 Neighborhoods Chain Code

Two consecutive pixel points define a segment. The direction of each segment is coded
by using a numbering scheme as shown in Figure 2.1. Suppose that \( \{d_0, d_1, \ldots, d_{N-1}\} \)
represents the chain codes of the contour \( \Phi \), where \( d_i \) is the direction of the segment
from point \( p_i \) to \( p_{i+1} \). A function \( G_\alpha(d_i), 0 \leq \alpha \leq 7 \), is defined for testing whether the
chain code \( d_i \) is in the \( \alpha \) direction.

Definition 2.4.1 Function \( G_\alpha(d_i) \) is defined as follow:

\[
G_\alpha(d_i) = \begin{cases} 
1 & \text{if } d_i = \alpha, \\
0 & \text{otherwise}, 
\end{cases}
\]  
(2.12)

2.4.2 Directional Flow Change

The following definitions describe how to create a directional flow-change function: \( \delta 
function” based on the chain codes, and how to find out the critical points based on the
\( \delta \) function.

Definition 2.4.2. The directional flow change at point \( p_i \in \Phi \) with the segment \( J \) on both
sides of \( p_i \) is defined as \( \delta(i,J) \)

\[
\delta(i,J) = \sum_{\alpha=0}^{7} |\delta_\alpha(i,J)|
\]  
(2.13)
where the flow-change is defined as:

\[
\delta_a(i,J) = \left| \sum_{k=i-j}^{i+j-1} G_a(d_k) - \sum_{k=i-j}^{i-1} G_a(d_k) \right|
\]  

(2.14)

\[\sum_{k=i-j}^{i+j-1} G_a(d_k)\] is the input flow in the \(\alpha\) direction of a contour segment with length \(J\) at point \(p_i \in \Phi\). \[\sum_{k=i-j}^{i+j-1} G_a(d_k)\] is the output flow in the \(\alpha\) direction of a contour segment with length \(J\) at point \(p_i \in \phi\).

**Definition 2.4.3** A critical point \(p_i \in \phi\) is the point that the directional flow of the contour at this point changes significantly and the following conditions should be satisfied:

1. let \(J = \gamma \times N, \; 0.01 \leq \gamma \leq 0.05\);

2. \(\delta(i,J) > t \times J\) for some \(t\) with \(0.6 \leq t\).

3. \(\delta(i,J) \geq \delta(k,J)\) for all \(k\) with \(i-L \leq k \leq i+L\), where \(L\) is an integer such that \(0.5 \times J \leq L \leq J\).

4. if \(p_k\) is the next critical point after \(p_i\), then there is at least one point \(p_j\) with \(i < j < k\) such that \(\delta(j,J) \leq \theta \times \delta(i,J)\), where \(0.5 < \theta < 1\).

5. Let \(p_i\) be the previous critical point before \(p_i\), \(p_{k'}\) be the first point after \(p_i\) so that \(\delta(k',J) \leq \theta \times \delta(i',J)\), \(p_k\) be the first point after \(p_i\) so that \(\delta(j,J) \leq \theta \times \delta(i,J)\), if there are \(m\) points \(\{p_{i_1}, \ldots, p_{i_m}\}\) between \(p_{k'}\) and \(p_k\) such that \(\delta(i_1,J) = \cdots = \delta(i_m,J)\), then \(i = (i_1 + i_m)/2\).

6. \(|i-k| \geq L\) if \(p_k\) is a critical point other than \(p_i\).
2.4.3 Shape Description

After getting the critical points, a polygon, which can be used to approximate the contour of the shape, can be constructed by connecting these critical points as vertices. Thus, every side length between the two neighboring critical points and the angle at each critical point can be calculated. In order to facilitate shape matching, the angles of the polygon are converted into a “angle string” based on a angle coding scheme shown in Table 2.1.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Angle Code</th>
<th>Angle</th>
<th>Angle Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-21</td>
<td>0</td>
<td>180-205</td>
<td>8</td>
</tr>
<tr>
<td>22-43</td>
<td>1</td>
<td>206-227</td>
<td>9</td>
</tr>
<tr>
<td>44-65</td>
<td>2</td>
<td>228-249</td>
<td>A</td>
</tr>
<tr>
<td>66-87</td>
<td>3</td>
<td>270-250</td>
<td>B</td>
</tr>
<tr>
<td>88-109</td>
<td>4</td>
<td>272-293</td>
<td>C</td>
</tr>
<tr>
<td>110-131</td>
<td>5</td>
<td>294-315</td>
<td>D</td>
</tr>
<tr>
<td>132-153</td>
<td>6</td>
<td>316-337</td>
<td>E</td>
</tr>
<tr>
<td>154-179</td>
<td>7</td>
<td>338-360</td>
<td>F</td>
</tr>
</tbody>
</table>

**Table 2.1 Angle Coding Scheme**

**Definition 2.4.4** Given any shape, a finite set of numbers can be generated by the modified method. Let \( A = \{A_1, \ldots, A_n\} \) and \( B = \{B_1, \ldots, B_m\} \) be the sets of shape numbers associated with shape \( S_A \) and shape \( S_B \), respectively. Then \( S_A \) and \( S_B \) are of the same shape if there exist some \( A_i \in A \) and some \( B_j \in B \) such that \( A_i = B_j \) and if the corresponding side lengths of the polygons, which are used to represent the contours the shapes, are also approximately the same.
Based on the Definition 2.4.4, we can compare two images to see if they are the same shape or not. Based on the critical points, we also can estimate the position deviation between the input image and the template image.

**Definition 2.4.5** for a given image, the position deviations along x direction and y direction are:

\[ X \text{ deviation} = \text{Average of critical point coordinates along x direction (input image)} \]
\[ - \text{Average of critical point coordinates along x direction (template)} \quad (2.15) \]

\[ Y \text{ deviation} = \text{Average of critical point coordinates along y direction (input image)} \]
\[ - \text{Average of critical point coordinates along y direction (template)} \quad (2.16) \]

### 2.5 Proportional Controller

![Block Diagram of Closed-Loop Control](image)

Figure 2.2 Block Diagram of Closed-Loop Control

In Figure 2.2, the feedback system provides the error signal \( e \) by subtracting the measured value \( x_0 \) from the desired value \( x \):

\[ e = x - x_0 \]

In a proportional control system, the error signal is amplified by an amplifier gain \( G \) to yield an output \( v_c \), which is the control signal:

\[ v_c = Ge \]
Chapter 3
Algorithm Design

3.1 Introduction

Image processing algorithms in machine vision typically include four steps as shown in Figure 3.1.

![Figure 3.1 Block Diagram of Image Processing Steps](image)

Image preprocessing step is to process the image to generate another one that containing only the desired image. Segmentation step is to separate the interested area from the rest part of the image. Feature extraction step is to extract the feature information from the segmented image. Image analysis step is to compare the extracted information with the known standards.

As introduced in Chapter 2, three methods can be employed for image analysis. In this Chapter, three algorithms based on statistical approach, cross correlation and DFC methods are discussed in details. In last section, detailed control strategy is presented.
3.2 Image Preprocessing

In industrial environment, there exist variety of noise sources. For a raw image coming from image grabber, noise may be included in it. This polluted image affects the reliability of the result.

3.2.1 3×3 Neighborhood Smoothing

3×3 neighborhood smoothing is one of the popular method in image processing. The operators transform an image by replacing each pixel with a value generated by averaging pixels in its neighborhood as shown in Figure 3.2.

<table>
<thead>
<tr>
<th>P_{i-1,j-1}</th>
<th>P_{i-1,j}</th>
<th>P_{i-1,j+1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_{i,j-1}</td>
<td>P_{i,j}</td>
<td>P_{i,j+1}</td>
</tr>
<tr>
<td>P_{i+1,j-1}</td>
<td>P_{i+1,j}</td>
<td>P_{i+1,j+1}</td>
</tr>
</tbody>
</table>

Figure 3.2 3×3 Neighborhood of Pixel P_{i,j}

The pixel value P_{i,j} can be replaced by the following:

\[ g(i, j) = \frac{1}{3 \times 3} \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} f(i, j); \]

(3.1)

This method can effectively reduce the noise from the image. But it may blur the edge of the image. It is used in statistical approach method and cross correlation method.

3.2.2 3×3 Median Filter

For DFC method, we have to use 3×3 median filter. Given a raw image, we can replace the pixel value p_{i,j} by the median pixel value p_{m,n} of its 3×3 neighborhood. For example, given an array \[ S = \begin{bmatrix} 20 & 50 & 60 \\ 30 & 70 & 80 \\ 90 & 50 & 40 \end{bmatrix}, \] the values can be sorted as
$A = \{20, 30, 40, 50, 60, 70, 80, 90\}$. The value 70 in the center of 2D array $S$ can be replaced by 50 the 5th value of array $A$.

### 3.2.3 Adaptive Threshold Method for Image Enhancement

Adaptive threshold method is based on the image’s gray intensities in some way. The purpose is to cope with variations in light and part appearance. The average value of the given $M \times N$ image $f(x, y)$ is:

$$\text{ave} = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y);$$

(3.2)

The input image can be transformed into:

$$f(x, y) = \begin{cases} 255 & \text{if } f(x, y) \geq k \times \text{ave}; \\ 0 & \text{if } f(x, y) \geq k \times \text{ave}, \end{cases} \quad k \text{ is a constant}$$

(3.3)

### 3.3 Segmentation

![Figure 3.3 (a) The Image of The Foam Barrier, (b) The Interested Image Area](image)

Figure 3.3 (a) The Image of The Foam Barrier, (b) The Interested Image Area
The image that we are interested and the whole image that we could get from the camera are illustrated in Figure 3.3.

3.3.1 The Existing Segmentation Method

![Figure 3.4 Criteria of Existing Segmentation Method](image)

The existing segmentation method is shown in Figure 3.4. It is based on the image with one whole foam barrier in the middle, part of another foam barrier in upper side and part of a foam barrier in down side. One of the obvious features of the image is that a narrow black band lies between every two neighborhoods foam barriers. In order to crop it into the same size as that of the template, the steps are as the following:

**Step 1.** Looking for the 1st Black band in the input image

Suppose \( G(i, j) \) represents the gray level of a pixel in the \( M \times N \) input image. The summation of the gray levels of all the pixels in each row is calculated.

\[
y(i) = \sum_{j=0}^{N-1} G(i, j)
\]

The maximum row value ‘MaxValue’ can be found according to the following equation.

\[
MaxValue = \max(y(i))
\]
If a certain $y(i)$ is less than or equal to $f \times MaxValue$, where $f$ is the constant, the corresponding row is called a 'Black Line'; otherwise, it is called a 'White Line'. If the 1st black line is found, and the succeeding 3 lines are still black lines, then the 1st black line is the 'Band0' shown in Figure 3.4.

**Step 2.** Looking for the 1st white band in the input image starting from top to bottom

'White Band' is defined as a band, which has at least 4 white lines out of 5 consecutive horizontal lines, Following a similar procedure as step 1, the 1st white band can be found, which is shown as 'Band1' shown in Figure 3.4. This band should be the approximate beginning line of the target foam barrier.

**Step 3.** Looking for the 2nd 'Black Band'

Using the same procedure as Step 1, the 2nd black band, which is shown as 'Band2' in Figure 3.4, can be found. This is the approximate end of the target foam barrier.

**Step 4.** Looking for the approximate 'Central Line' of the target foam barrier.

The center hole of the foam barrier is almost between the Band1 and Band2. So, the central line, which represents the reference position of the target foam barrier. The position can be obtained by $v_{central} = (Band1 + Band2) / 2$.

**Step 5.** Using the same procedures as above steps. Such as, finding the first black band, the first white band and the second black band from left to the right. We can find the horizontal central line $h_{central}$, which can be view as the horizontal reference point of the foam barrier.

**Step 6.** Based on the size of the template $M \times M$ and the reference position $(h_{central}, v_{central})$ got from above steps, we can crop the image as the following:

$$topline = v_{central} - M / 2;$$
$$bottoline = v_{central} + M / 2 - 1;$$
$$leftline = h_{central} - M / 2;$$
$$rightline = h_{central} + M / 2 - 1;$$
Because the time consumption of the cross correlation is increased greatly as the size of the cropped image is increased a little, this segmentation method is only used in cross correlation algorithm to crop the image.

There are two drawbacks with this segmentation method.

- Comparing with only one foam barrier in the 320×240 image, we can not get higher resolution with this method because it involves three foam barrier in a fixed size image (320×240).
- It is very difficult to handle the special cases when we implement it. For example, one of the special cases is shown in Figure 3.5

![Figure 3.5 Special Case](image)

3.3.2 New Segmentation Method

![Figure 3.6 1D Histogram Segmentation Method](image)
The new segmentation method is illustrated in Figure 3.6. First we zoom in the image to fill the whole range of 320×240 frame with only one foam barrier. Then follow the steps described as below to crop the image to get the interested part.

**Step 1.** Looking for first black line ‘Firstxbline’

Using Equation 2.1 to construct the 1D histogram function in x direction. Generally, the function looks like the one shown in Figure 3.7. In some cases, there maybe is only one blank area or no blank area in 1D histogram.

![Diagram showing the general 1D histogram function of foam barrier in x direction.](image)

**Figure 3.7 The General 1D Histogram Function of Foam Barrier in X Direction**

Now, we can find the MaxValue of the histogram. \( \text{MaxValue} = \max(H(x)) \); then compare each value of \( H(x) \) to find the ‘firstxbline’ starting from left to right. The first position that the \( H(m) \) is less than the threshold becomes the ‘firstxbline’.

\[
\text{firstxbline} = m; \quad \text{if} \quad H(m) \leq f \times \text{MaxValue(threshold)} \quad f \text{ is a constant}
\]

**Step 2.** Looking for the last black line ‘lastxbline’

Using the same method as the Step 1 but starting from right to left, we can get the ‘lastxbline’.

**Step 3.** Crop the image along the ‘firstxbline’ and the ‘lastxbline’ in x direction, we can get the area between the ‘firstxbline’ and the ‘lastxbline’.

**Step 4.** Based on the cropped image in last step, the 1D histogram function \( H(y) \) in y direction can be constructed by using Equation 2.2. Repeat Step 1, but starting from top.
to bottom to find the ‘firstybline’ in y direction. Repeat Step 2, but starting from bottom
to top to find the ‘lastybline’ in y direction.

Step 5. Crop the image along the ‘firstybline’ and the ‘lastybline’ in y direction, then we
can get the needed image. The cropped image is shown in Figure 3.8.

Figure 3.8 Cropped Image of The Foam Barrier

(a)  (b)

Figure 3.9 Special Cases for New Segmentation Method

Two special cases are shown in Figure 3.9. For case (a), we can use the bottom line as the
‘lastybline’. For case (b), we can use the top line as the ‘firstybline’.

This new segmentation method improves the resolution greatly because we can zoom in
the image as long as we can get enough information and it’s very easy to deal with the
special cases. This segmentation method is employed in DFC and statistical approach.

3.4 Feature Extraction and Image Analysis

After getting the simplified image that only contains the necessary information, we can
extract the feature from the image and conduct image analysis. Three methods have been
introduced in Chapter 2. In this section, they will be used to solve the problems.

3.4.1 Statistical Approach

![Image of a histogram and mean value graph]

**Figure 3.10 Statistical Approach Based Image Analysis**

During the image segmentation process, two 1D histogram functions \( H(x), H(y) \) have been constructed in x and y direction, respectively, and ‘firstxbline’, ‘lastxbline’, ‘firstybline’ and ‘lastybline’ have been found. Now, we can use Equation 2.3 and 2.4 to calculate the mean values in x and y direction, respectively.

\[
x_c = \frac{\sum_{x=\text{firstxline}}^{\text{lastxline}-1} xH(x)}{\sum_{x=\text{firstxline}}^{\text{lastxline}-1} H(x)} ; \quad y_c = \frac{\sum_{y=\text{firstyline}}^{\text{lastyline}-1} yH(y)}{\sum_{y=\text{firstyline}}^{\text{lastyline}-1} H(y)} ;
\]

(3.6)

The mean values \( x_c \) and \( y_c \) can be treated as the center point of the foam barrier in x and y direction, respectively. Comparing with the standard center position \( X_{sc} \) and \( Y_{sc} \), we can get the position deviation.
\[ x_{dev} = x_c - X_{sc}, \quad y_{dev} = y_c - Y_{sc}; \] (3.7)

After getting the mean values, we can estimate the width and the height of the foam barrier by using Equation 2.5 and 2.6.

\[
W_x = f_x \sqrt{\frac{\sum_{x=\text{first line}}^{\text{last line}-1} (x-x_c)^2 H(x)}{\sum_{x=\text{first line}}^{\text{last line}} H(x)}}, \quad H_y = f_y \sqrt{\frac{\sum_{y=\text{first line}}^{\text{last line}-1} (y-y_c)^2 H(y)}{\sum_{y=\text{first line}}^{\text{last line}} H(y)}};
\] (3.8)

where \( f_x \) and \( f_y \) are experiment values. \( W_x \) and \( H_y \) can be viewed as the width and height of the foam barrier in pixels. Comparing with the standard width \( W \) and height \( H \), we can estimate if the stretching behavior occurs.

\[
X_{stretch} = W_x - W; \quad Y_{stretch} = H_y - H;
\] (3.9)

![Figure 3.11 Stretched Foam Barrier](image)

Let us give an example. Suppose that image shown in Figure 3.10 represents the standard template image,
After image segmentation, we can get the following values:

\[
\begin{align*}
\text{firstxbl} & = 0 \quad ; \quad \text{lastxbl} = 319 \quad ; \\
\text{firstybl} & = 0 \quad ; \quad \text{lastybl} = 239 \quad ; \\
\end{align*}
\]

Then, we can calculate the center position by using Equation 3.6 and get:

\[
X_{sc} = 141, \quad \text{and} \quad Y_{sc} = 134;
\]

Using Equation 3.8, we can get the estimated width and height:

\[
W = 54, \quad H = 63;
\]

Suppose that image shown in Figure 3.11 represents the stretched image. Using the same way as the standard one, we can get the following data:

\[
\begin{align*}
x_c &= 140, \quad \text{and} \quad y_c = 122 \\
W_x &= 50, \quad H_y = 67;
\end{align*}
\]

Using Equation 3.7, we can get the position deviation:

\[
x_{dev} = -1, \quad y_{dev} = -12;
\]

Using Equation 3.9, we can estimate the stretching behavior:

\[
X_{stretch} = -4, \quad Y_{stretch} = 4;
\]

### 3.4.2 Method Based on Cross Correlation

Based on the method described in Section 3.3.1, we can crop the input image to the same size as that of the template image. After cropping the image, we can get a fix size of image and the reference position \((x_c, y_c)\). In this work, the size is \(110 \times 110\) because it is the smallest size that includes all necessary information of the foam barrier.

There are two ways to prevent false correlation peaks caused by the changes in the mean image gray level. One is normalized cross correlation that has already been introduced in
Section 3.3.1. Another way is to do edge enhancement first, then use two edged image (cropped input image and template image) to do the cross correlation analysis. In this research work, edge enhancement method is employed. Refer to [5] for details. Using Equation 2.7, we can calculate the correlation between the edged template and the edged input image. The computation method of correlation is illustrated in Figure 3.12.

![Figure 3.12 Cross Correlation Between Two Images](image)

After cross correlation, we can exam the position of the peak value. If the peak value equals to the peak value of the auto correlation of the template image, that means no stretching behavior occurs. The position deviation is:

\[ x_{dev} = X_{w} - x_{c}, \quad y_{dev} = Y_{w} - y_{c}; \]

### 3.4.3 Directional Flow Change Method (DFC)

![Figure 3.13 (a) Cropped Image (b) Enlarged Image](image)
A special cases based on the new image segmentation method described in Section 3.3.2 is illustrated in Figure 3.13 (a). Suppose that each grid represents one pixel of the image, the white area is the image of the object and the shaded area is the background. We can see that part of the object is out of the scope. For the cases like this, we cannot use the 8-neighborhood method as we introduced in Chapter 2 for chain coding. So, we have to modify it by adding one more pixels along the edges of the cropped image. An enlarged image is shown in Figure 3.13 (b).

After enlarging the binary image, we can trace the contour of the object by using 8-neighborhood method. Refer to [5] for details.

The whole contour starting from top left corner of the contour can be traced and the position array of the pixels on the contour can be got.

After getting chain code of the contour, $\delta$ function can be built and the critical points can be found based on the Definitions 2.4.1, 2.4.2 and 2.4.3 described in Chapter 2. The steps are:

**Step 1.** Set a factor $\gamma$ between 0.01 and 0.05. Let $J = \gamma \times N$, where $N$ is the total numbers of the pixels on the contour.

![Diagram](image)

Figure 3.14 Input Flow and Output Flow of The Pixel, $J=4$
**Step 2.** As shown in Figure 3.14, using equation 2.12 to get the direction functions $G_a(d_i)$ of the input segment of the contour with length $J$ and output segment of the contour with length $J$ at the point $p_i \in \Phi$.

**Step 3.** Calculate the input flow $\sum_{k=i-J}^{i-1} G_a(d_k)$ and output flow $\sum_{k=i}^{i+J} G_a(d_k)$ based on Definition 2.4.2. Then, use Equation 2.13, 2.14 to get the directional flow-change function $\delta(i, J)$.

![Graph](image)

Figure 3.15 $\delta$ Function of The Cropped Foam-Barrier Image

**Step 4.** Smooth average of the $\delta$ function:

$$\delta(i, J) = \frac{\delta(i-A, J) + \cdots + \delta(i, J) + \cdots + \delta(i+A-1, J)}{2A}$$

**Step 5.** Choose the values for $\theta$, $t$ and $L$ so that $0.5 \leq \theta \leq 1$, $t \geq 0.6$ and $0.5J \leq L \leq J$.

**Step 6.** Critical points = 0;

**Step 7.** Starting from $p_a$, search for the first two points $p_p$ and $p_k$ with $p < k$ so that $\delta(p, J) \geq t \times J$; $\delta(j, J) \leq \delta(p, J)$, for all $0 \leq j \leq k$ and $\delta(k, J) \leq \theta \times \delta(p, J)$,
Step 8. \( i = k, d = k; \)

Step 9. Increase \( i \) during searching for the first point \( p_{d'} \) and all points \( p_{j'} \) with \( j' < d' \) so that \( \delta(p_{j'}, J) \geq \delta(j, J) \), for all \( j \)'s with \( d \leq j \leq d' \); \( \delta(p_{j'}, J) > t \times J \) and \( \delta(d, J) \leq \theta \times \delta(p_{j'}, J) \), At every new point \( p_i \), test if \( (i \mod N) = k \), if true, output critical point and terminate.

Step 10. If more than one \( p_{j'} \) points are found between \( p_d \) and \( p_{d'} \) say \( \{p_1, \cdots, p_m\} \), then the critical point candidate \( = (i + m)/2 \).

If \text{critical points} = 0, then

(a) Add \( p_{\text{critical}} \) to critical points;

(b) \text{Previous} = \text{candidate};

(c) \( d = d' \); Go to step 9.

Step 11. If \( \text{candidate} - \text{previous} \geq L \), then

(a) Add \( p_{\text{previous}} \) to critical points;

(b) \text{Previous} = \text{candidate};

Else if \( \delta(\text{previous}, J) < \delta(\text{candidate}, J) \), then

\( \text{Previous} = \text{candidate} \).

Step 12. \( d = d' \); Go to step 9.

Figure 3.16 (a) Original Image, (b) Constructed Polygon of The Foam Barrier
Through the 12 Steps, we can find the critical points and construct the polygon to approximate the original shape based on the method described in Section 2.4.3.

Now the side length and angles of the polygon can be calculated. The angle coding scheme "Table 2.1" is too sensitive to be used in this design. So, the table should be modified according to the real angles of the polygon to ensure that the angles of the polygon are in the middle of the angle ranges not just at the cutoff position. Table 3.1 is a modified angle coding scheme that is more suitable to foam barrier.

Table 3.1 Modified Angle Coding Scheme

<table>
<thead>
<tr>
<th>Angle</th>
<th>Angle Code</th>
<th>Angle</th>
<th>Angle Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-29</td>
<td>0</td>
<td>200-209</td>
<td>8</td>
</tr>
<tr>
<td>30-39</td>
<td>1</td>
<td>210-239</td>
<td>9</td>
</tr>
<tr>
<td>40-69</td>
<td>2</td>
<td>240-249</td>
<td>A</td>
</tr>
<tr>
<td>70-79</td>
<td>3</td>
<td>250-279</td>
<td>B</td>
</tr>
<tr>
<td>80-109</td>
<td>4</td>
<td>280-289</td>
<td>C</td>
</tr>
<tr>
<td>110-119</td>
<td>5</td>
<td>290-319</td>
<td>D</td>
</tr>
<tr>
<td>120-149</td>
<td>6</td>
<td>320-329</td>
<td>E</td>
</tr>
<tr>
<td>150-159</td>
<td>7</td>
<td>330-359</td>
<td>F</td>
</tr>
</tbody>
</table>

Based on the Table 3.1, a angle string can be obtained. The angle string of the polygon shown in Figure 3.16 (b) is "29546558", and the side lengths are: 34, 85, 79, 84, 52, 45, 63, 99 pixels. This data is the feature of the foam barrier. According to Definition 2.4.4, we can compare this data with the feature of the template to see if they are the same pattern. In order to keep the result stable and accurate, a boundary is applied when comparing side lengths. If the calculated side length $L_c$ is within the boundary of the corresponding side length $L_i$ of the template, then the calculated one equals to the template one.
\[ L_d = L_t \quad \text{if} \quad 98\% \times L_t \leq L_c \leq 102\% \times L_t \]

Using Equation 2.15, 2.16, we can calculate the position deviation.

### 3.5 Review of The Three Algorithms

In this section, three flowchart will be presented to outline the algorithms.

#### 3.5.1 The Flowchart of The Statistical Approach

![Flowchart of The Statistical Approach Algorithm](image)

Figure 3.17 Flowchart of The Statistical Approach Algorithm
3.5.2 The Flowchart of The Algorithm Based on Cross Correlation

![Flowchart](image)

**Figure 3.18 Flowchart of The Algorithm Based on Cross Correlation**

3.5.3 The Flowchart of Directional Flow-Change Method

![Flowchart](image)

Algorithm Design 31
3.6 Control Strategy

After image analysis, the position deviation is obtained in pixels. It needs to be converted
to “millimeter” by:

\[ m\text{deviation} = f \times p\text{deviation}; \]

where, \( m\text{deviation} \) is the position deviation in millimeter, \( p\text{deviation} \) is the position deviation in pixels, \( f \) is the factor that is related the “zoom” of the camera and the height of the camera.

Consider that servomotors will be used in this design. The relationship of the position deviation \( m\text{deviation} \), time \( t \) and the speed of servomotor \( sp \) is:

\[ m\text{deviation} = t \times sp; \]

This is a linear relationship. As introduced in Section 2.5, the proportional control strategy can be employed in this design.

![Figure 3.20 Block Diagram of The Proportional Control](image)

Figure 3.20 Block Diagram of The Proportional Control
Chapter 4

Hardware Implementation

4.1 Introduction

In this chapter, the hardware architecture and the principle will be discussed. Then the basic models will be introduced individually. Finally, the specification and feature of the system is also reviewed. As introduced in Section 1.2, hardware consists of three units: vision, computer and tracking. Vision unit includes CCD camera and image grabber, tracking unit includes control unit and servo motors.

4.2 Lighting

Since the image of the object is generated by a image sensor that converts the optical signal reflected by the object into the electrical signal. Lighting plays a very important role in machine vision system. In order to reduce the inference from the ambient source, enhance the contrast and to simplify image processing, a dedicated light should be applied in a machine vision system.

According to the light source, there are three types of light sources: point, diffuse and collimated. Each type of light source can create different effect. For a shape recognition application, a diffuse light source will create better effect [19].
There are variety of practical light source available: LED array, Fiber-Optic array, arc lamp, strobe tube, fluorescent tube, Quartz Halogen bulbs, incandescent light bulb and laser etc. Each one has different advantages and disadvantages. One has to make a selection for specific applications.

In this application, two long and straight fluorescent tubes are employed as light sources because fluorescent tube is inexpensive, easy to diffuse and easy to mount, as well as it can generate white, efficient and uniform distribution light. The drawback is that the light pulses at 120Hz Frequency (in north America). But this can be neglect in this research because the size of the object is large enough. The size of the single foam barrier is illustrated in Figure 4.2.

Figure 4.1 Block Diagram of The Light Source

Figure 4.2 Size of the Foam Barrier
4.3 Lens

The mechanism of imaging is shown in Figure 4.3. We can see that the focal length decides the size of the image. Lenses with a focal length of more than 25mm are telephoto lenses. These lenses make the object appear larger than it really is and show less of the normal scene. Lenses with a focal length shorter than 15mm are wide angle lenses. They can make the object appear smaller than the standard 20mm lens.

![Figure 4.3 Imaging Principle of Thin Lens](image)

In this application, a 35mm focal length lens is chosen. The working distance is set to 28cm and the field of view is set to 105×80 mm². The maximum resolution we can get is 0.33 mm. Besides the resolution and the size of the image, another factor that affects the image quality is distortion. Two kinds of distortion exists for a monochromatic image: (a) pincushion distortion, (b) barrel distortion. The two types of distortions are shown in Figure 4.4. This is a very important factor when we design an accurate position tracking system. We have to compensate the distortion for position tracking. As shown in the experiment results (Chapter 6), some errors between the calculated deviations and the real deviations could be caused by this factor.
Figure 4.4 Lens Distortion: (a) Pincushion Distortion; (b) Barrel Distortion

4.4 CCD Camera

Charge coupled device—CCD is the mainstream product in current image sensor market.

The working principle of the CCD sensor is illustrated in Figure 4.5.

Figure 4.5 Interlaced Transfer CCD Sensor
First, it converts light signal into an electronic charge at discrete sites. Then it transfers the packets of charge within the silicon substrate. After that, it converts the charge to voltage and amplifies the signal. Lastly, it outputs the signal.

According to the output signal, there are two types of industry cameras in the market—digital cameras and analog cameras. Actually, a digital camera can be thought of as an analog camera with an embedded A/D converter. So, it can output digital data that can be directly transmitted into a computer without a frame grabber, and provide higher resolution images than analog images. The drawback is that it is more expensive than analog cameras. Analog cameras include progressive cameras, interlaced cameras and line scan cameras. Progressive cameras and line scan cameras are suitable for capturing the image of moving objects. In the assembly line of automotive door handle escutcheons, because the static image can be captured, so we choose interlaced CCTV cameras as the image sensor.

Conventional CCTV cameras are widely applied in machine vision systems. In North America, all interlaced CCTV cameras are based on EIA RS-170 standard rules that each frame has 525 lines, scanning speed is 30 frames a second, field rate is 60 per second. In this research, Everfocus EX100 camera is employed as the image sensor. The specifications of the camera are as the following:

- B/W,
- Interlace
- 30 frames/s
- 640×480 pixels/frame
- Composite or S-video output
- EIA RS-170
4.5 DT3120 Image Grabber

DT3120 image grabber is a high quality, low-cost PCI frame grabber designed for industrial, scientific and medical imaging applications [21][22][23]. The DT3120 enables the acquisition of a standard monochrome, composite color or S-video color image from a single camera. The imaging software SDK enables us to set up the image grabber, to develop imaging applications. Because of the PCI Bus Master design and intelligent scatter/gather memory management architecture, the DT3120 frame grabbers handle large amounts of image data quickly and effectively, and without CPU intervention. This leaves the host CPU free to do other tasks such as image processing, data manipulation or other operations. The block diagram of DT3120 is illustrated in Figure 4.6.

Figure 4.6 Block Diagram of DT3120 Image Grabber
The DT3120 image grabber accepts one composite (CVSB) or one S-video input source at a time by connecting the composite input to J2 terminal or connecting the S-video input to J3 terminal on the DT3120 board (Refer to Figure 4.7). Using software, we can adjust brightness and contrast of the DT3120 board to affect the intensity of the gray level. The pixel clock of DT3120 is 12.5MHz for 60Hz image formats, and the pixels are available to the DT3120 in the increments of the Pixel Period, which is equal to 1/clock frequency.

DT3120 board can accept one of the following trigger sources:

- A software trigger- Default mode, the board can acquire a frame when a software command is issued.

- An external trigger- By attaching a digital signal to connector J1 on the board, the frame acquisition can be synchronized with external events.
The incoming external trigger is forwarded to the device driver as an interrupt. The interrupt is processed by the device driver to start the acquisition if desired. It can be specified whether to start image acquisition when the board detects either a rising-edge transition or a falling-edge transition.

The DT3120 board can detect the horizontal and vertical synchronous signals, as well as odd and even fields, and digitize the incoming video signal from the video input signal. Horizontal synchronous pulse is asserted low for 4μs typically. Vertical synchronous pulses are asserted low for 230μs typically. The odd field indication changes state on falling edge of the vertical synchronous signal.

The total video area is a complete set of horizontal and vertical input lines from which the active video area and the frame can be extracted. It includes all parts of the signal, including non-visual portions. The active video area floats in the total video area. It is defined as the part of the incoming signal that contains valid video data. The components of a single horizontal line of video is illustrated in Figure 4.8. A horizontal line of video is identified by the falling edge of the horizontal synchronization, and a field is composed of a collection of horizontal lines defined by the active line count. Pixel measurements are relative to the horizontal reference point, which is defined as the beginning of the horizontal synchronization. Each field of video also contains blanking information and lines of active video. The components of a single vertical field of non-interlaced video is shown in Figure 4.9. Line measurements are relative to the vertical reference point, which is defined as the beginning of the vertical synchronization. Lines are measured in terms of pixels. Both the vertical and horizontal video signal settings are fixed and cannot be programmed.
Figure 4.8 Horizontal Video Signal

The frame is the portion of the active video area that we want to digitize. This is the region in which we are interested. The frame size, frame type, frame scaling and frame storage modes are directly related to our applications.

Figure 4.9 Vertical Video Signal
The default frame size setting for the DT3120 is 640×480 pixels. The definition of the frame size and the relationships of the top, left, height and width of the frame is illustrated in Figure 4.10.

![Figure 4.10 Spatial Relationship of Video Signal](image)

DT3120 board can acquire interlaced frames. The video signal is defined as two consecutive fields - even field that contains lines 0,2,4, ... and odd field that contains lines 1,3,5, and so on. Using software, one of the following types of frame acquisitions can be selected:

- Interlaced frames, starting on next even field (default),
- Interlaced frames, starting on next odd field.
- Interlaced frames, starting on next field.

The gray level image data can be stored in monochrome format-8 bits per pixel, as shown in Figure 4.11.
DT3120 board has two types of image acquisition mode: passthru operation and single capture operation. Passthru operation means that the DT3120 can continuously captures and displays video data until you stop the operation. Usually, this mode can be employed to focus and position the camera. Single capture mode means that the DT3120 capture the still image and display. The source origin of an image is the upper left corner of the image. On the DT3120 board, the source origin is always 0,0. Using DT3120, a single full frame, a single field, multi full frames or multiple fields can be acquired. Data is stored to an area in system memory that is allocated by DT3120 device driver. The minimum memory for monochrome image is 307200 bytes [21][22][23].

4.6 Motorola 68HC912B32 Microcontroller

The MC68HC912B32 microcontroller unit is a high speed 16 bits device with standard on chip peripherals and 32k bytes flash EEPROM, 1k bytes RAM and 768 bytes EEPROM memory [24]. In single chip mode, it has 64 I/O pin available for designer. In this design, we only need one serial communication interface (SCI) for communication
with PC and two output compare pin for generating pulses to control two servo motors. A MC68HC912B32 working in single chip mode is suitable for this research.

![Memory Map Diagram]

**Figure 4.12 MC68HC912B32 Single Chip Mode Memory Map**

The memory map for single chip mode of operation immediately after reset is illustrated in Figure 4.12. After reset, the internal register block is located at $0000-$01FF, the register-following space is at $0200-$03FF, and RAM is at $0800-$0BFF, EEPROM is located at $0D00-$0FFF, the Flash EEPROM array is located from address $8000 to $FFFF in single chip mode. The Flash EEPROM module requires an external program/erase voltage ($V_{FP}$) to program or erase the Flash EEPROM array. $V_{FP}$ is provided via an external $V_{FP}$ pin.

CPU12 exceptions include resets and interrupts. Each exception has an associated 16-bit vector that points to the memory location. Vectors are stored in the upper 128 bytes of the standard 64-kbyte address map from $FFD0-$FFFF. In this research, four exceptions are
employed, which are one non-maskable RESET ( $FFE-$FFFF), three maskable
interrupts-asynchronous serial communication SCI0 ($FFD6-$FFD7), timer channel2
($FFEA-$FFEB), timer channel3 ($FFE8-$FFE9).

The standard timer module consists of 16-bit software-programmable counter driven by
a pre-scaler. It contains eight complete 16-bit input capture/output compare channels and
one 16-bit pulse accumulator. In this research, two of them are used to generate pulses to
drive servomotors.

![Serial Interface Block Diagram](image)

Figure 4.13 Serial Interface Block Diagram

The serial communication interface (SCI) is a non-return zero format (one start, eight or
nine data, and one stop bit) asynchronous communication protocol with independent
internal baud rate generation circuit and an SCI transmitter and receiver. This system is
compatible with standard RS232 systems. It can be configured for eight or nine data bits
(including one odd or even parity bit). If enabled, parity is generated in hardware for
transmitted and received data. It has some control registers to set up the Baud Rate,
transmitter/receiver operation mode etc. Two port pins (PS0,PS1) provide the external
interface for the transmitted data (TxD) and the received data (RxD).

**4.7 Tower Hobbies TS-69 S3K Servo Motor**

Because of the high precision, accuracy and small size, servomotors have been widely

Hardware Implementation 46
applied in robotic position. In this design, two Tower Hobbies TS-69 S3K servomotors are employed. It has powerful motor and molded gear train. The specifications of the servomotor are as the following:

- Speed: .20sec/60 degrees
- Torque: 42oz-in
- Weight: 1.61oz
- Size: 1.59”×0.77”×1.41”

Three wires are connected to the servomotor: the red one is for 4.8v power supply, the black one is for ground and the white one is for control.

![Diagram](image)

Figure 4.14 Control Signal and Modifying Servo Motors

The control signal and modified the servomotors are illustrated in Figure 4.14. The control signal for the servomotor is 50Hz pulse and the width of the pulse decides which direction the motor will rotate, clockwise or counterclockwise. If the pulse width is 1.50ms, the servomotor will keep at the neutral position. If the pulse width is less than 1.50ms, the motor will rotate counter clock wisely and vice versa. But the inner drive circuit and the plastic gear limit the servomotor to rotate over ±90° from the neutral point.
In order to make it rotate continuously, the circuit and the gear need to be modified. Two 2.3 kΩ resistors are used to replace the feedback potentiometer and the stop bar of the plastic gear has to be cut off.

4.8 Rail Rack for Tracking

A two-dimensional rail rack is built for supporting the camera and tracking the position deviation. The whole rail rack is made out of aluminum. It has two rails in orthogonal direction, one for x direction tracking, another one for y direction tracking. The pitch of the driving axis is 0.8mm. The picture of the rail rack is shown in Figure 4.15.

Figure 4.15 Picture of The Rail Rack
4.9 Hardware Implementation of The System

The block diagram of the system architecture is shown in Figure 4.16. CCD camera converts the optical signal reflected by the object into analog voltage signal. Then ADC converts the analog signal into digital signal. Through the look up table (LUT), 8-bit gray level image signal (monochrome image) is generated. After that, we can get the required frame size of image via image scaling and cropping control unit. Last, the formatted image data is stored into the PC’s system memory through first in first out (FIFO) and PCI interface. As mentioned in DT3120 image grabber section, we can configure the color intensity, trigger mode, frame size, frame type, frame storage mode and image acquisition mode via software. The data transfer rate of PCI interface is 132 M-byte/s, and a video camera that sends 30frames/s produces data streams of 10 to 40 MB/s. So, an unlimited number of consecutive frames can be transferred across the bus in real time.
The core of the system is the PC. The minimum requirements for the PC are:

- A Pentium MMX® CPU.
- At least 800M hard disk and 64M bytes RAM.
- At least one free PCI bus slot and RS232 serial port.
- At least 640×480 VGA monitor.
- Windows98/NT operating system.

The PC is for storing the image data, processing the image, making decision and displaying image and information. The position deviation signal is sent to microcontroller via RS232. Control strategy is carried out by microcontroller. The schematic diagram of the servo motor control is shown in Figure 4.17. Max232 chip is employed to convert signals between 5v NRZ signal and RS232 signal. 74LS04 chip is the buffer to drive the servomotors.

![Schematic Diagram of Servo Motor Control](image)

**Figure 4.17 Schematic of Servo Motor Control**
Chapter 5

Software Implementation

5.1 Introduction

The software design is based on Windows98/NT/2000 operating system. Microsoft® VC++6.0 is one of the most popular tool. It can take advantage of Microsoft Foundation Class Library (MFC) to interactive with Windows OS and optimize the user code automatically. In this design, Object-Oriented Design (OOD) method is employed. DT3120 SDK package is included for setting up and controlling the image grabber. In order to speed up software programming, Microsoft® Vision SDK1.2 is used for image operation. The embedded software is developed in C.

5.2 DT3120 SDK

The 32-bit frame grabber SDK provides user with over 50 functions for DT PCI frame grabbers, which include general imaging, camera control, input control, memory management, passthru and overlay [23].

The functions allow users to perform the following general system operations:

- Initializing a frame grabber board
- Determining a frame grabber board’s capabilities
- Defining timeouts
• Working with status codes

• Releasing a frame grabber board

Camera control functions provide users with the abilities to set up the following specifications of the video input signal, which are necessary to this research.

• Video signal type

• Video input channel

• External trigger

• Video area and frame size

Memory management functions enable users to allocate frame buffers to store image data. Passthru operation functions allow users to carry out the following operations:

• Performing a passthru operation

• Adjusting the source origin

• Scaling the passthru image

• Taking a snapshot

• Creating overlays

5.3 Microsoft® Vision SDK1.2

Microsoft Vision SDK is a library for writing programs to perform image manipulation and analysis on computers running Microsoft Windows operating systems [25]. It includes classes and functions for working with images. It is suitable for fast, real-time image processing. The current version of SDK is intended for use with Microsoft VC++6.0. Three of the projects in SDK1.2 are used this research. The descriptions of the three projects are:
The **Wizard** project adds the Vision AppWizard to the Project tab of the File/New dialog in Microsoft Visual Studio. In this work, this AppWizard is used to create new projects that include Vision SDK functions.

- The **VisCore** project defines the CVisImage and CVisSequence classes that are used to work with both single image and sequences of images. It also defines classes for exceptions and reference-counted memory that used in other projects.

- The **VisDisplay** project defines the CVisPane and CVisPaneArray classes that are used to create windows to display and interact with a single image or a group of images.

The projects and their classes employed in this research is illustrated in Figure 5.1.

<table>
<thead>
<tr>
<th>VisCore</th>
<th>VisDisplay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard pixel types</td>
<td>Display an Image</td>
</tr>
<tr>
<td>Image class</td>
<td></td>
</tr>
<tr>
<td>Image File I/O</td>
<td></td>
</tr>
<tr>
<td>Exceptions</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.1 The Related Projects and The Classes in SDK

The CVisImage class is the core of VisSDK. Similar to the Windows’s bitmap header the CVisImage stores a variety of properties about an image and a pointer to the memory. The Vision SDK uses the standard MFC RECT structure to define the rectangular area of the image. The structure shown in Figure 5.2 defines TOP, BOTTOM, LEFT and RIGHT coordinates such that the data area is within the columns from and including LEFT over to, but not including RIGHT, and the rows from and including TOP down to, but not including BOTTOM.

Software Implementation 53
A pixel is addressed by x and y coordinates, where x is the column and y is the row, with the coordinates increasing from left to right and from top to bottom. The origin of an image can be placed anywhere. The SDK provides a lot of standard pixel types. In this work, CVIs(Gray)BytePixel is used, which is an 8 bits “unsigned char” data type.

5.4 Software Design

Software design involves graphic user interface (GUI), and hardware interface.
All application classes in this design are shown in Figure 5.5. Based on Microsoft MFC, five basic application classes can be generated. They are Documentation class, Main Routine class, Windows Frame class, Display class and About Dialog class. These five classes construct the main stream routine of the application and the basic Windows Graphic User Interface (GUI). As shown in Figure 5.4, the basic application GUI has a window frame, menu area, button area and client area. All operations can be carried out through menu or buttons. The client area is used to display the image, the position deviation and pattern information.

![Image of Basic Windows Interface]

**Figure 5.4 Basic Windows Interface**

The flow chart of the main stream of the application is illustrated in Figure 5.5. This is an event-drive application. Refer to Appendix D for detailed operation instructions. Besides the classes derived from Microsoft Vision SDK1.2, nine classes are designed based on different objects, which involve hardware configuration, image acquisition, image processing and analysis.
Figure 5.5 Flowchart of The Main Routine

All classes and subroutines are described as the follows:

- CsmartVisionApp: This is main routine class for maintaining the association
among the document, the frame, and the display classes, and for dispatching the commands and messages from Windows.

- CsmartVisionDoc: This document class is a container of the application data
- CMainFrame: This is the windows frame class for generating the frame for window interface.
- CSmartVisionView: This display class is for showing the information on the window interface.
- CSerialPortDlg and CSerialPort: These two classes are designed for serial port configuration dialog and serial data transfer. The default configuration is “COM1, 9600, 8, None, 1, 10”. The RS232 subroutine is shown in Figure 5.6. The GUI of RS232 configuration dialog is illustrated in Figure 5.7.

![Diagram]

Figure 5.6 RS232 Configuration Subroutine
CImgGrabber1: This class includes all operation functions of the image grabber, such as opening and closing image grabber, configuration, passthru operation and single image acquisition, etc.

CFoamBarrier: This class is specially designed for the foam barrier material. It describes the feature of the foam barrier and includes some image processing functions and the functions of cropping the foam barrier image. The Statistic Approach algorithm is also included in this class.

CCrossCorrelation, CDirectionalFlowChange and CEdgeEnhancement: These are the algorithms for image analysis and pattern recognition. These classes are based on gray level image, and the CCrossCorrelation is designed for the image of 110×110 pixels specifically.

CGrabTempDlg: This dialog class provides the operator with an interface to get the template image, choose the algorithm and setup the timer period and resolution. The GUI of the “Grab template dialog” is shown in Figure 5.8. The flowchart of the “Get image template subroutine” is shown in Figure 5.9.
Figure 5.8 Grab Image Template Dialog

Figure 5.9 Flowchart of The "Grab Template Dialog"
As shown in Figure 5.10, the timer triggers the single-frame acquisition event. After getting the image of the object, image processing algorithms are executed. The timer function \texttt{UINT SetTimer(UINT nIDEvent, UINT nElapse)}”, the call back function \texttt{OnTimer(UINT nIDEvent)}” and \texttt{KillTimer(m_nTimer)}” control the whole subroutine.
Figure 5.11 Flowchart of Image Acquisition Subroutine
The process of image acquisition and the operation of the image grabber is illustrated in Figure 5.11. The core of the software implementation has been introduced in Chapter 2 and Chapter 3. In order to make the software running smoothly, some exceptions and error handling are added. (see Appendix A)

As mentioned before, the application will run in the Demo routine if no DT 3120 image grabber is detected. The differences between the Demo and the normal routine are that the Demo read the image from hard drive instead of image grabber and it does not have positioning function. Except for that, all others are the same as the normal routine.

5.5 Embedded Software Design

The embedded control software for MC68HC912B32 microcontroller is written in C language. After compiling by ICC12 cross compiler, S19 codes can be generated, which can be downloaded to Motorola’s microcontrollers.

The SCIhandler() interrupt function is designed to receive the data from RS232. The buffer array Fifo[8] is for holding the data from PC. The format is as follows:

<table>
<thead>
<tr>
<th>Y sign</th>
<th>Y ones</th>
<th>Y tens</th>
<th>Y hundreds</th>
<th>X sign</th>
<th>X ones</th>
<th>X tens</th>
<th>X hundreds</th>
</tr>
</thead>
</table>

Figure 5.12 Data Format of Communication Between PC and Microcontroller

For example, if the position deviation is “x=-20, y=+135”, the transmitted data is “+531-020”. TOC2handler() and TOC3handler() functions are designed for timer interruptions to generate the 50 Hz (T=20ms) or 100Hz (T=10ms) pulses to drive the servo motors. The flowchart of the position tracking system is shown in Figure 5.13.
Figure 5.13 Flowchart of Embedded Control
Figure 5.14 Flowchart of SCI Interrupt

Figure 5.15 Flowchart of TOC Interrupt
Chapter 6

Experiments and results

6.1 Introduction

The developed "Vision Tracking System" can be used to test, verify and improve the algorithms. This system can potentially be applied to solving the existing industrial problem, such as the position deviation and the stretching behavior of the foam barrier in the foam barrier assembly line for automotive door handle escutcheons. In this Chapter, some experiments are introduced first. Then the results are discussed and compared. Finally, the analysis of the algorithms' computational complexity and the memory consumption is given.

6.2 Experiments with The Statistical Approach

The statistical approach method is a popular tool to estimate the position and the size of the 2D objects.

Experiment 1.

Given a small piece of white square paper shown in Figure 6.1, the size is $31 \times 27 \text{ mm}^2$. Because the height is very small, it can be thought as a planar object.

Step 1.3 $\times 3$ neighborhood smoothing.
Figure 6.1 The Image of The White Square Paper

**Step 2.** Adaptive threshold. For this experiment with two 30W florescent tubes light, the best threshold value is 200. The different effect with different threshold is illustrated in Figure 6.2.

![Threshold Comparison](image)

(a) threshold=220  
(b) threshold=200  
(c) threshold=180

Figure 6.2 Effectiveness of Image Enhancement with Different Thresholds

**Step 3.** Construct 1D histograms. The 1D histograms $H(x)$ and $H(y)$ are shown in the Figure 6.3.

![Histograms](image)

(a) $H(x)$ of the square image  
(b) $H(y)$ of the square image

Figure 6.3 The 1D Histograms $H(x)$ and $H(y)$ of The Square Image
**Step 4.** Calculate the mean values of $H(x)$ and $H(y)$. Get the center position $x_c$ and $y_c$.

$$x_c = X_{\text{mean}} = 149, \quad y_c = Y_{\text{mean}} = 131;$$

**Step 5.** Calculate the standard deviation to get the size $w$ and $h$;

$$w = 27, \quad h = 31;$$

**Experiment 2.**

This experiment is based on a white bottle cap with 28mm diameter and 15mm height. Because this is a cubic object, the calculated size will be a little bit different from the real size. It is caused by the angle between the camera and the object.

The image of the bottle cap after enhancement is shown in Figure 6.4.

![Figure 6.4 Enhanced Image of The Bottle Cap, Threshold=150](image)

Figure 6.4 Enhanced Image of The Bottle Cap, Threshold=150

(a) $H(x)$ of The Bottle Cap  \hspace{1cm} (b) $H(y)$ of The Bottle Cap

Figure 6.5 1D Histograms of The Bottle Cap in x and y Direction
The 1D histogram of the bottle cap, $H(x)$ and $H(y)$ is shown in Figure 6.5.

The center position is: $x_c = 161$, $y_c = 123$

The diameters in $x$, $y$ direction are: $D_x = 29$mm, $D_y = 28$mm

We can see that the diameter has 1mm error in $x$ direction, no error in $y$ direction because there is a larger angle $\theta$ in $x$ direction than the smaller angle $r$ in $y$ direction between the cap and the camera. The errors will be different as the position of the cap changes. The angles between the cap and the camera are illustrated in Figure 6.6.

![Figure 6.6 Angles Between The Camera and The Cubic Object](image)

**Experiment 3.**

As shown in Figure 3.10 and Figure 3.11, this experiment shows how to estimate the stretching behavior based on the changes of the size.

**Experiment 4.**

This experiment is for position deviation detection based on the statistic approach. The purpose is to test the robustness and accuracy of the algorithm. The result of the position deviation without tracking unit is listed in Table 6.1.
### Table 6.1 Comparison of The Position Deviation

<table>
<thead>
<tr>
<th>real deviation</th>
<th>estimated deviation</th>
<th>real deviation</th>
<th>estimated deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xr Yr Xc Yc</td>
<td>Xr Yr Xc Yc</td>
<td>Xr Yr Xc Yc</td>
<td>Xr Yr Xc Yc</td>
</tr>
<tr>
<td>-1 1 -1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>-2 2 -2 2</td>
<td>2 2 2 2</td>
<td>2 2 2 2</td>
<td>2 2 2 2</td>
</tr>
<tr>
<td>-3 3 -3 3</td>
<td>3 3 3 3</td>
<td>3 3 3 3</td>
<td>3 3 3 3</td>
</tr>
<tr>
<td>-4 4 -4 4</td>
<td>4 4 4 4</td>
<td>4 4 4 4</td>
<td>4 4 4 4</td>
</tr>
<tr>
<td>-5 5 -5 5</td>
<td>5 5 5 5</td>
<td>5 5 5 5</td>
<td>5 5 5 5</td>
</tr>
<tr>
<td>-6 6 -6 6</td>
<td>6 6 6 6</td>
<td>6 6 6 6</td>
<td>6 6 6 6</td>
</tr>
<tr>
<td>-7 7 -7 7</td>
<td>7 7 7 7</td>
<td>7 7 7 7</td>
<td>7 7 7 7</td>
</tr>
<tr>
<td>-8 8 -8 8</td>
<td>8 8 8 8</td>
<td>8 8 8 8</td>
<td>8 8 8 8</td>
</tr>
<tr>
<td>-9 9 -9 9</td>
<td>9 9 9 9</td>
<td>9 9 9 9</td>
<td>9 9 9 9</td>
</tr>
<tr>
<td>-10 10 -10 10</td>
<td>10 10 10 11</td>
<td>10 10 11 11</td>
<td>10 10 11 11</td>
</tr>
<tr>
<td>-12 12 -12 12</td>
<td>12 12 12 13</td>
<td>12 12 13 13</td>
<td>12 12 13 13</td>
</tr>
<tr>
<td>-13 13 -14 14</td>
<td>13 13 14 14</td>
<td>13 13 14 14</td>
<td>13 13 14 14</td>
</tr>
<tr>
<td>-14 14 -15 15</td>
<td>14 14 15 15</td>
<td>14 14 15 15</td>
<td>14 14 15 15</td>
</tr>
<tr>
<td>-15 15 -16 15</td>
<td>15 15 15 17</td>
<td>15 15 17 17</td>
<td>15 15 17 17</td>
</tr>
<tr>
<td>-1 1 -1 1</td>
<td>-1 1 -1 1</td>
<td>-1 1 -1 1</td>
<td>-1 1 -1 1</td>
</tr>
<tr>
<td>-2 2 -2 2</td>
<td>-2 2 -2 2</td>
<td>-2 2 -2 2</td>
<td>-2 2 -2 2</td>
</tr>
<tr>
<td>-3 3 -3 3</td>
<td>-3 3 -3 3</td>
<td>-3 3 -3 3</td>
<td>-3 3 -3 3</td>
</tr>
<tr>
<td>-4 4 -4 4</td>
<td>-4 4 -4 4</td>
<td>-4 4 -4 4</td>
<td>-4 4 -4 4</td>
</tr>
<tr>
<td>-5 5 -5 5</td>
<td>-5 5 -5 5</td>
<td>-5 5 -5 5</td>
<td>-5 5 -5 5</td>
</tr>
<tr>
<td>-6 6 -6 6</td>
<td>-6 6 -6 7</td>
<td>-6 6 -6 7</td>
<td>-6 6 -6 7</td>
</tr>
<tr>
<td>-7 7 -7 8</td>
<td>-7 8 -7 8</td>
<td>-7 8 -7 8</td>
<td>-7 8 -7 8</td>
</tr>
<tr>
<td>-8 8 -8 9</td>
<td>-8 9 -8 8</td>
<td>-8 9 -8 8</td>
<td>-8 9 -8 8</td>
</tr>
<tr>
<td>-10 10 -11 10</td>
<td>-10 10 -11 11</td>
<td>-10 10 -11 11</td>
<td>-10 10 -11 11</td>
</tr>
<tr>
<td>-12 12 -14 12</td>
<td>-12 14 -12 12</td>
<td>-12 14 -12 12</td>
<td>-12 14 -12 12</td>
</tr>
<tr>
<td>-13 13 -13 13</td>
<td>-13 bottom 13</td>
<td>-13 13 14 -13</td>
<td>-13 13 14 -13</td>
</tr>
<tr>
<td>-14 14 -14 14</td>
<td>14 bottom 14</td>
<td>-14 14 bottom</td>
<td>-14 14 bottom</td>
</tr>
</tbody>
</table>

**unit:** mm
Figure 6.7 Relationship Between The Real Deviations and Estimated Deviations

This experiment shows:
estimated deviation = real deviation for $|\Delta x| \leq 5, |\Delta y| \leq 5$;

estimated deviation \equiv real deviation for $5 < |\Delta x| \leq 12, 5 < |\Delta y| \leq 12$;

6.3 Experiments with The Cross Correlation Based Algorithm

Experiment 1. Position deviation detection

Table 6.2 Comparison of Position Deviation in 1st Quadrant

<table>
<thead>
<tr>
<th>real deviation</th>
<th>estimated deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xr  Yr</td>
<td>Xc  Yc</td>
</tr>
<tr>
<td>1   1</td>
<td>2    1</td>
</tr>
<tr>
<td>2   2</td>
<td>3    2</td>
</tr>
<tr>
<td>3   3</td>
<td>4    3</td>
</tr>
<tr>
<td>4   4</td>
<td>5    4</td>
</tr>
<tr>
<td>5   5</td>
<td>6    4,5,6</td>
</tr>
<tr>
<td>6   6</td>
<td>8    6</td>
</tr>
<tr>
<td>7   7</td>
<td>10,9 10,9</td>
</tr>
<tr>
<td>8   8</td>
<td>11   10,11</td>
</tr>
<tr>
<td>9   9</td>
<td>12   12,11</td>
</tr>
<tr>
<td>10  10</td>
<td>14   14</td>
</tr>
<tr>
<td>11  11</td>
<td>15   15</td>
</tr>
<tr>
<td>12  12</td>
<td>16   17</td>
</tr>
<tr>
<td>13  13</td>
<td>17   18</td>
</tr>
</tbody>
</table>

The data of position deviations in first quadrant is listed. We can see that some errors exceed the tolerance (1mm), and the some results is not stable. For example, the real deviation $Y_r$ is 5 but the calculated deviation $Y_c$ varies from 4 to 6.

Experiment 2. Stretching behavior analysis (Refer to Figure 3.10, 3.11)

For a given template, the peak value of the auto-correlation is $P_a = 15423014$. For an input image of the stretched foam barrier, the peak value of the cross correlation is
\[ P_c = 14038126, \text{ which is less than the threshold } t = P_A \times 0.95. \text{ This is a stretched object.} \]

6.4 Experiments with DFC Algorithm

The DFC algorithm is designed for position deviation analysis and stretching behavior analysis. The following experiments are to test this algorithm.

**Experiment 1.**

A standard rectangle and its enhanced image are shown in Figure 6.8.

(a) Original Standard Rectangle Image  
(b) Enhanced Image  
Threshold = 200

Figure 6.8 Input Images Before Chain Coding

(a)  
(b)  

Figure 6.9 (a) \( \delta \) Function of The Image, (b) Constructed Polygon
The $\delta$ function of the standard rectangle is shown in Figure 6.9 (a). It has 4 distinguish peaks that are greater than 20. They can be thought as 4 critical points. The constructed polygon with the 4 critical points is illustrated in Figure 6.9 (b).

**Experiment 2.**

As shown in Figure 6.10, it is the same rectangle as the one in Experiment 1, but the upper-left edge is rough.

![Figure 6.10 The Image of The Rectangle with One Rough Edge](image)

From the $\delta$ function (Figure 6.11 (a)), we can see that there are some small peaks in the last segment from 250-321, which represents the rough part of the upper-left edge of the rectangle. By setting the threshold properly, we can get the same result as experiment 1.
As shown in Figure 6.11 (b), the constructed polygon based on the 4 critical points getting from the DFC analysis of the $\delta$ function is the same as the result in Experiment 1.

**Experiment 3.**

This experiment is for stretching behavior analysis with DFC method. Based on the constructed polygon, we can compare the input image with the template. A given foam barrier is shown in Figure 6.12.

![Figure 6.12 The Image of The Foam Barrier](image1)

![Figure 6.13 The $\delta$ Function of The Foam Barrier](image2)
After 3×3 median filtering and image enhancement (threshold=130), we can use 8
neighborhood method to trace the contour and get $\delta$ function (Figure 6.13) of the foam
barrier. By using DFC method, 9 critical points can be got and a polygon (Figure 6.14)
representing the foam barrier can be constructed.

![Polygon of The Foam Barrier](image)

Figure 6.14 Constructed Polygon of The Foam Barrier

The angle string is:

"2A6656558",

and the side lengths are:

"23, 82, 72, 81, 48, 44, 55, 65, 36"

The template’s angle string is

"2A6656558"

and the side lengths are:

"22, 82, 74, 81, 47, 44, 55, 65, 36"

Compare the two pattern, they can be treated as the same pattern. Therefore, there is no
stretching behavior occurring.

A different $\delta$ function of the same object under the same condition is illustrated in
Figure 6.15 \( \delta \) Function of The Foam Barrier

The 9th peak value in Figure 6.13 changes. It become very difficult to tell this change from other small peaks, such as the peak at 127 position. Now there are only 8 critical points. The constructed polygon is illustrated in Figure 6.16

Figure 6.16 Constructed Polygon

The angle string is "2A546558" and the side lengths are "41, 84, 73, 80, 47, 45, 54, 82". This results in a totally different pattern, which suggests a stretched product or other pattern.
Experiment 4.

This is a repetitive test with a circle shape bottle cap shown in Figure 6.17.

![Figure 6.17 The Image of The Bottle Cap](image)

We repeat three times with DFC method under the same condition and with no position deviation. The three calculated $\delta$ functions are shown in Figure 6.18.

![Figure 6.18 Three Calculated $\delta$ Functions of The Bottle Cap](image)
From the $\delta$ functions, we can see that some of the peak values are changed every time.

We cannot get the reliable critical points. The constructed polygons are shown in Figure 6.19.

![Three constructed polygons of the bottle cap](image)

Figure 6.19 Three Constructed Polygons of The Bottle Cap

<table>
<thead>
<tr>
<th></th>
<th>1st capture</th>
<th>2nd capture</th>
<th>3rd capture</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Critical</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td><strong>Points</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Side Lengths</strong></td>
<td>8, 37, 40, 19, 28, 29, 31, 94</td>
<td>32, 40, 29, 17, 64, 29, 73, 21</td>
<td>26, 26, 64, 50, 15, 41, 47, 35</td>
</tr>
<tr>
<td><strong>Angle String</strong></td>
<td>456666666</td>
<td>555566666</td>
<td>556655666</td>
</tr>
</tbody>
</table>

$r = 0.03$, $t = 0.5$;

The corresponding side lengths and angle string of the polygons is listed in Table 6.3. The results are different from each other.

This experiment proves that we cannot get expected results by using DFC method to do the shape recognition analysis with the circle shape objects.

**Experiment 5.**

This experiment is for position deviation detection. The experiment data is listed in Table 6.4.
Table 6.4 Comparison of The Position Deviation

<table>
<thead>
<tr>
<th>real deviation</th>
<th>calculated deviation</th>
<th>real deviation</th>
<th>calculated deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xr Yr Xc Yc</td>
<td></td>
<td>Xr Yr Xc Yc</td>
<td></td>
</tr>
<tr>
<td>-1 1</td>
<td>-1 1</td>
<td>1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>-2 2</td>
<td>-2 2</td>
<td>2 2</td>
<td>2 2</td>
</tr>
<tr>
<td>-3 3</td>
<td>-3 3</td>
<td>3 3</td>
<td>3 3</td>
</tr>
<tr>
<td>-4 4</td>
<td>-4 4</td>
<td>4 4</td>
<td>4 4</td>
</tr>
<tr>
<td>-5 5</td>
<td>-5 5</td>
<td>5 5</td>
<td>5 5</td>
</tr>
<tr>
<td>-6 6</td>
<td>-6 6</td>
<td>6 6</td>
<td>6 6</td>
</tr>
<tr>
<td>-7 7</td>
<td>-7 7</td>
<td>7 7</td>
<td>8 7</td>
</tr>
<tr>
<td>-8 8</td>
<td>-8 8</td>
<td>8 8</td>
<td>9 9</td>
</tr>
<tr>
<td>-9 9</td>
<td>-9 9</td>
<td>9 9</td>
<td>10 9</td>
</tr>
<tr>
<td>-10 10</td>
<td>-10 10</td>
<td>10 10</td>
<td>11 11</td>
</tr>
<tr>
<td>-11 11</td>
<td>-11 11</td>
<td>11 11</td>
<td>12 12</td>
</tr>
<tr>
<td>-12 12</td>
<td>-12 12</td>
<td>12 12</td>
<td>13 13</td>
</tr>
<tr>
<td>-13 13</td>
<td>-13 13</td>
<td>13 13</td>
<td>15 14</td>
</tr>
<tr>
<td>-14 14</td>
<td>-14 14</td>
<td>14 14</td>
<td>16 15</td>
</tr>
<tr>
<td>-15 15</td>
<td>-15 16</td>
<td>15 15</td>
<td></td>
</tr>
</tbody>
</table>
Figure 6.20 Relationship Between Real Deviation and Estimated Deviation
From Table 6.4 and the Figure 6.20, we can see that DFC algorithm has an excellent performance on position deviation detection.

### 6.5 Experiment with Tracking System

This experiment is to test the tracking system and control strategy. Experiment 6.3 shows that the cross correlation based algorithm is not suitable for position deviation detection. Therefore, this experiment is based on statistical approach and DFC algorithms. Because the rail pitch is 0.8mm, the ratio between moving distance and the revolution is 5:4. So, we modified the measured position deviation by multiply the ratio 1.2. As shown in Table 6.5, the desired result 0s can be obtained by employing proportional control strategy.

<table>
<thead>
<tr>
<th>real deviation</th>
<th>DFC tracking</th>
<th>statistics tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Xr Yr Xd Yd</td>
<td>Xa Ya</td>
</tr>
<tr>
<td>-5 -5</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>5 -5</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>-5 5</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>5 5</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 4</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>5 3</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>-5 -3</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>7 0</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>-7 0</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>-4 -10</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>4 10</td>
<td>0 0 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

*Unit: mm*

| Experiments and Results | 81 |
6.6 Algorithm Analysis and Comparison

In this section, the algorithms will be analyzed and compared based on the experiments. The computational complexity of the algorithms is also discussed.

6.6.1 Algorithm Based on Statistical Approach

From the experiment results in Section 6.2, we can see that this algorithm can be used for position deviation and stretching behavior analysis of planar object, no matter what kind of the shape the object is. It can be used to estimate the size of changes in x and y direction quantitatively. The drawback is that it can not recognize the pattern of the objects.

The computation of this algorithm is fixed as long as the frame size is fixed (320×240 in this design) and the computational cost can be estimated exactly. For example, for a given 320×240 preprocessed image, the computational cost is 153,600 additions for getting \( H(x) \) and \( H(y) \), 1120 additions and 560 multiplications for getting center position, and 1120 additions and 1120 multiplications for getting estimated size. For a 433Mhz CPU, the time cost is only several milliseconds. Obviously, the memory consumption is also small. It only needs \( 2 \times 320 \times 240 \) bytes memory.

This algorithm is a robust and accurate real time algorithm that can be implemented with micro-controllers or DSPs.

6.6.2 Algorithm Based on Cross Correlation

The experiment in Section 6.3 shows that the proposed cross correlation method is not suitable for the manufacturing problem because the result is not stable and accurate. The proposed algorithm uses a simple segmentation method to find the position deviation and employ cross correlation method to find out if stretching behavior occurs. We can get
robust and accurate result on stretching behavior analysis but can not get reliable result on position deviation analysis.

Because of the convolution calculation, the correlation algorithm consumes large amount of time. For the foam barrier, the size of the cropped image is the same size as the template (110x110). The cross correlation cost the 433MHz CPU almost 15s. After calculation, a 219x219 size array is generated. So, this algorithm runs very slowly and needs large memory.

6.6.3 Algorithm Based on DFC

The experiment in Section 6.4 shows that this algorithm can be used for position deviation detection and stretching behavior analysis. Especially, it has best performance for the planar objects with sharpen angles. But, because the foam barrier has some round angles and rough edges, there is a problem in pattern recognition. As discussed before, the peak values of the $\delta$ function for the round angles are not distinguished. Because of this, the numbers of critical points could be changed so that the constructed pattern may not match the template.

The computation of this algorithm is also simple because only the pixels on the contour need to be processed. No complex calculation exists. For a 433MHz CPU, It only costs hundreds of milliseconds to get result.

6.6.4 Comparison of Three Algorithm

Each algorithm has its own advantages and drawbacks, and has different applications. Based on the results in this research, we can draw a conclusion that the statistical approach algorithm is best for solving foam barrier problem existing in the assembly line. It works fast, robust and accurate. The comparison of the three algorithms is listed in Table 6.6.
<table>
<thead>
<tr>
<th></th>
<th>statistics</th>
<th>cross correlation</th>
<th>DFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>position</td>
<td>accuracy</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>deviation</td>
<td>robustness</td>
<td>good</td>
<td>bad</td>
</tr>
<tr>
<td>analysis</td>
<td>resolution</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>stretching</td>
<td>quantity</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>behavior</td>
<td>robustness</td>
<td>good</td>
<td>good</td>
</tr>
<tr>
<td>analysis</td>
<td>direction</td>
<td>only x, y direction</td>
<td>any</td>
</tr>
<tr>
<td></td>
<td>speed</td>
<td>milliseconds</td>
<td>seconds</td>
</tr>
<tr>
<td></td>
<td>memory</td>
<td>small</td>
<td>large</td>
</tr>
<tr>
<td></td>
<td>limitation</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusion and Future Work

7.1 Conclusion

Three aspects are presented in this thesis. They are:

1. Development and implementation of the algorithms.
2. Hardware and software design.
3. Experiments and the analysis of the algorithms

The whole system can be applied to solving the position deviation detection problem caused by stretching behavior of the foam barrier material in the assembly line for automotive door handle escutcheons. It can also be used as a simulation platform for the development of new algorithms and for testing and verifying other algorithms. Because OOD method is used in software design and hardware modules are use’ in hardware design, this system provides us with high flexibility. It can be easily expended to 3D system.

The hardware design involves optics, electronics and mechanics, which include lighting source, lens, CCD camera, image grabber, computer, micro-controller, servomotors and 2D rail rack. Each part has a significant influence on the speed, robustness and accuracy of the system. In addition, another significant feature of this system is low cost.
The software design involves GUI, multi-thread, SDK, Hardware configuration, RS232 and exception etc. The software package provides operators with friendly interface and simple instructions. Embedded software for micro-controller is written in C. So, it is easy to understand and easy to be modified.

The algorithm design involves image processing and pattern recognition. Image processing procedure also plays an important role in the performance of the system. In order to get reliable result, different image processing methods are applied to different algorithms.

The contributions of this research are summarized as follows:

1. The low cost 2D vision tracking system is developed for the applications to 2 dimensional position deviation tracking.

2. A fast, efficient, robust and accurate algorithm based on statistic approach method is developed and implemented to find position deviation caused by stretching behavior of the objects.

3. The existing algorithm based on the correlation analysis method is modified, implemented, tested and verified.

4. The existing algorithm based on the Directional Flow-Change method is modified, implemented, tested and verified. Specific image preprocessing is developed to improve the efficiency and robustness of the algorithm.

5. For the application to the foam barrier problem, a dedicated segmentation method is designed to improve the resolution of the algorithms. The angle-coding scheme is also modified.
7.2 Comments on Three Algorithms

The algorithm based on statistical approach method is efficient for high-speed 2D position tracking and stretching behavior analysis. For the stretching behavior analysis, it can get the quantity of the size change in x and y direction. But it cannot be applied for size analysis in other directions and for pattern recognition. This algorithm can be easily implemented and embedded into microprocessor/FPGA based smart vision sensor.

The algorithm based on the cross correlation analysis method has been proved to be unreliable for position deviation detection. The method and the result for position deviation analysis are not reliable. The resolution is limited because of the computation complexity of the cross correlation method. But the normalized cross correlation method can be widely applied in pattern recognition. It can tell if the stretching behavior occurs no matter which direction it is. This algorithm can be easily implemented but need high-speed device and large amounts of memory.

The algorithm based on the Directional Flow-Change method is efficient for position deviation analysis. For stretching behavior analysis or pattern recognition, it is suitable for the objects with sharpen angle shapes but not for the objects with round angle and rough edges.

7.3 Future Work

In this thesis, the algorithms design is based on the 2D spatial domain. We need to develop new 3D algorithms to solve the position deviation problem. The vision tracking system can also be extended to be a 3D system by adding relevant components.

The image capture of the system is based on stationary image caption. For the
applications to moving objects, the camera need to be changed to progressive camera.

And the special image processing methods have to be added to eliminate the problems caused by moving.

As the progress of modern electronic technology, new powerful microprocessors, DSPs and FPGAs are becoming popular in industrial applications. Research has to be conducted to embedded high efficient algorithms into small size, low power consumption and flexible smart vision sensor systems.
References


[20] ADAPT-912 Starter Package, Technological Arts Inc.,


[26] www.datatranslation.com


[28] www.imagecraft.com

[29] www.machinevisiononline.org/public/articals/


[31] www.xilinx.com
Appendix B

VC++ Source Code

Appendix B from Page 93 to 207 is on the attached Floppy Disk.
Appendix C

C Code for MC68HC912B32

Appendix C from Page 208 to 214 is on the attached Floppy Disk.
Appendix D

Operation Manual

Figure 1. System Connection Diagram

A: Rear panel of the computer
B: Front panel of the Tri-Power Supply
C: Front panel of the microcontroller box
D: Rear panel of the microcontroller box
E: The up side of the CCD camera
F: Servo motor X
G: Servo motor Y
Chapter 1. Installation

1.1 System requirement
- IBM compatible computer-Pentium 433 CPU or above
- At least 64MB RAM and 40MB hard disk space
- At least one PCI slot and one RS232 series port
- Windows9X/NT/2000/XP operating system

1.2 Software Installation
- Install DT3120 device driver software
- Install the SmartVision software
  1. Insert “SmartVision” floppy disk 1 into the floppy disk drive.
  2. Double click “setup” icon.
  3. Follow the instructions to finish the installation.

1.3 Hardware Installation
Refer to the Figure 1.
- Put the rail rack on a stable table
- Adjust the camera to the lowest position of the rail rack
- Turn off the computer
- Open the cover of the computer
- Plug the DT3120 image grabber card in a PCI slot
- Close the cover
- Connect the BNC cable to the middle terminal of the DT3120 card, connect the other end of the BNC cable to the camera
- Connect the RS232 cable to the computer, connect the other end of the RS232 cable to the microcontroller.
- Connect the camera to the 12v DC power supply
- Connect the two florence tube to the 120v AC power supply
- Connect the servo motors to the 5v DC power supply, two red wires to the positive output port, two black wires to the ground.
- Connect the white signal wires of the servo motors to the output port of the microcontroller. White wire with sticker to the red output port with sticker, the other wire to the other red output port.
- Connect the ground output port of the 5v DC to the ground port of the microcontroller

1.4 Download the program to the 68HC11 microcontroller
- Before using the control unit, the battery should be full charged at least 14 hours
- Connect the RS232 cable between the microcontroller and PC
- Open the cover of the control box
- Press down the stop button and hold, turn on the switch.
- After two seconds, release the stop button
• Run the download software on the computer—"double click the ‘shortcut to hbdl’ icon", a dialog "handy board download" appears on the desktop.
• Choose the correct COM port, make sure the "set config register=0x0C".
• Click the "download" button, in the "open" dialog window, select the "twowhole.s19" in a specified folder. Then click "OK" button. It begin to download the codes to the microcontroller.
• Wait a couple of minutes until the computer finish downloading.
• Turn off the microcontroller, close the cover, then turn on the microcontroller
• It is ready to work!

Chapter 2. Operation
• Turn on the AC power supply to the light
• Turn on the DC power supply to the camera and servo motors
• Turn on the control unit
• Turn on the PC
• Double click the icon "shortcut to SmartVision" on the desktop, a window "SmartVision" appears.

A: Menu area        D: Object missing information
B: Button area      E: Position deviation in x direction
C: Client area      F: Position deviation in y direction
1: "New" button to open a new file
2: Open a BMP image file from disks
   
   /*
   Save an image file in BMP format. Now, the function is disabled  */
3: Start button to start timer triggered image capture and analysis
4: Stop button to stop the image capture and analysis.
5: Template capture button
6: Positioning button
7: Stop positioning button
8: Reset button

- Click positioning button “6”, the video of the objects appear on the client area of the application Window.
- Adjust the position of the objects, and the focus of the camera to make the objects on the correct position and the correct size
- Click the template capture button “5”. Grab Template” dialog appears.

In the Grab Template dialog:

1. Click Grab button in “Grab Template” dialog, the image of the template appears on the preview area.
2. Choose the algorithms from the algorithm box.
3. Enter the Timer period in the Time box
4. Enter the resolution value in the resolution box
5. Click OK button to return to the application Window.

- Click positioning button “7” to stop the position mode
- Click the start button “3” to start image capture and image analysis
- If want to exist the running program, click the stop button “4” to stop the program.
- After stop the program, We can click “Reset” button “8” to clear the application Window.

Chapter 3 RS232 configuration and test

- Click “Setup-serial port” menu, a “Serial Port” dialog appear

In the Serial Port dialog:
1. Select the correct Serial port.
2. Select the correct baud rate, data bits, charity, stop bits, buffer value. The default values are “9600,8, None,1,10”
3. Click OK button to return the application Window

- Click “Send-Send” menu to test, if the servo motor moves than the system works.
Chapter 4. Software Interface

4.1. Introduction
SmartVision application software is an open platform, User can add their own image processing code into it by following the below instructions. Please read the following manuals first.

- The vision tracking system operation manual.
- DT frame grabber 32 SDK
- The Microsoft Vision SDK manual
- DT3120 user manual
- Jun Yang’s Thesis

4.2. If you want to add your own image processing code into this application source code without changing anything. Please pay more attention on the following:

- Built your own C++ classes with the combination of class type “CvisByteImage your image”. Refer to The Microsoft Vision SDK manual.
- Use the functions that “Vision SDK” provides for reading and analyzing and displaying images.
- Import your classes into “SmartVision” source code. Build instances of your classes in the class “CSmartVisionDoc”
- Pass the pointer of the image in “CSmartVisionDoc” class to your own classes by using the following method

    YourInstance.yourimage=m_image;

    * m_image hold the image data getting from the camera

- Make some changes in the function

```cpp
    void CSmartVisionDoc::OnImageGrabtemp()
    {
        ..... 
        add your codes
        ..... 
    }
```

- Make some changes in the function

```cpp
    BOOL CSmartVisionDoc::ProcessingImage()
    {
        ..... 
        add your codes
        ..... 
    }
```

- Make some changes in “CgrabTempDialog” class
- Make sure you already put all the necessary header files in your classes.
- Test and debug the changed SmartVision.
4.3 If you want to get ride of SmartVision platform and write your own codes, and wants to take advantage of the hardware interface classes that already exists in "SmartVision", you just import the following classes into your own source code

- CserialPort
- CserialPortDialog // for RS232 port communication
  Check my thesis to see the format of the data that will be transmitted to microcontroller.
- CImageGrabber1 // for image grabber configuration and operation

Read the class documentation to see how the functions in the classes work!
Before to build and run your source code, make sure you already add the include files and library to your "TOOLs-Options-Directories" of VC++ and add your link files to the "Project-Setting-Link"
Vita Auctoris

Jun Yang was born in October 24th, 1966 in Tianjin, China. He got his Bachelor of Engineering degree from the Department of Precision Instrumentation Engineering in Tianjin University in 1989. He is currently a Master Candidate of Applied Science in the Department of Electrical and Computer Engineering at the University of Windsor and hopes to graduate in summer 2002.