A PRECISION ROBOT ARM CONTROL FOR POSE CORRECTION ON PCB USING IMAGE PROCESSING TECHNIQUE

TASSANAI BANLUE

A THESIS SUBMITTED IN PARTIAL FULLFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2016

KMITL-2016- EN-D-018-015

การควบคุมแขนหุ่นยนต์แม่นยำสูงด้วยเทคนิคการประมวลผลภาพเพื่อแก้ไขตำแหน่งปลายแขนให้เข้าสู่เป้าหมายบนแผงวงจรอิเล็กทรอนิคส์

A PRECISION ROBOT ARM CONTROL FOR POSE CORRECTION ON PCB USING IMAGE PROCESSING TECHNIQUE

ทัศนัย บรรลือ

TASSANAI BANLUE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2559

KMITL-2016-EN-D-018-015

A PRECISION ROBOT ARM CONTROL FOR POSE CORRECTION ON PCB USING IMAGE PROCESSING TECHNIQUE

TASSANAI BANLUE

A THESIS SUBMITTED IN PARTIAL FULLFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2016

KMITL-2016- EN-D-018-015

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การควบคุมแขนหุ่นยนต์แม่นยำสูงด้วยเทคนิคการประมวลภาพเพื่อแก้ไขตำแหน่งปลายแขนให้เข้าสู่เป้าหมายบนแผงวงจรอิเล็กทรอนิคส์

Thesis Title A Precision Robot Arm Control for Pose Correction on PCB Using Image Processing Technique

นักศึกษา นายทัศนัย บรรลือ

รหัสประจำตัว 51060045

ปริญญา วิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.สุทธิชัย นพนาคีพงษ์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม รศ.ดร.ปิติเขต สู้รักษา

หมายเลขวิทยานิพนธ์ KMITL-2016-EN-D-018-015

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.ไกรสิน	ส่งวัฒนา	
ผศ.ดร.ศรวัฒน์	ชิวปรีชา	
ผศ.ดร.สุภาวดี	อร่ามวิทย์	
รศ.ดร.ยุทธพงษ์	รังสรรค์เสรี	
ผศ.ดร.สุทธิชัย	นพนาคีพงษ์	

วัน / เดือน/ ปี ที่สอบ วันพฤหัสบดีที่ 18 กุมภาพันธ์ พ.ศ. 2559 เวลา 13.00-15.00 น.
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร. คมสัน มาลีสี)
คณบดี คณะวิศวกรรมศาสตร์
วันที่ 18 กุมภาพันธ์ พ.ศ. 2559

หัวข้อวิทยานิพนธ์	การควบคุมแขนหุ่นยนต์แม่นยำสูงด้วยเทคนิคการประมวลผลภาพเพื่อแก้ไขตำแหน่งปลายแขนให้เข้าสู่เป้าหมายบนแผงวงจรอิเล็กทรอนิคส์
นักศึกษา	นายทัศนัย บรรลือ
รหัสประจำตัว	51060045
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2559
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.สุทธิชัย นพนาคีพงษ์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รศ.ดร.ปิติเขต สู้รักษา

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอการออกแบบและการนำไปใช้ของระบบการป้อนกลับข้อมูลภาพ ระบบการป้อนกลับข้อมูลจากภาพนี้ได้ถูกออกแบบเพื่อแก้ไขความคลาดเคลื่อนทางตำแหน่งของปลายแขนหุ่นยนต์ซึ่งประยุกต์ใช้ในโรงงานอุตสาหกรรม ปัจจัยหลายประการที่เกิดจากการใช้งานจริงในโรงงานอุตสาหกรรมก็มีส่วนต่อการออกแบบและการนำไปใช้งานด้วย เป็นที่ยอมรับทั่วไปว่าความผิดพลาดทางตำแหน่งของการวางปลายแขนนั้นสามารถเกิดขึ้นได้จากจุดหมุนที่เชื่อมต่อเข้าด้วยกันหลายจุดบนแขนเกร็งของหุ่นยนต์ ความผิดพลาดทางการหมุน การทำต้นแบบของระยะทางกายภาพ รวมถึงการกำหนดระยะตกกระทบของกล้อง ดังนั้นจึงเป็นสิ่งจำเป็นที่จะตรวจจับและป้อนกลับข้อมูลความผิดพลาดนี้ไปยังหุ่นยนต์ งานแขนกลหรือหุ่นยนต์อัตโนมัติส่วนใหญ่ทำงานเหมือนกันในเรื่องการเคลื่อนอุปกรณ์สัมผัส

ชิ้นงานที่ติดกับปลายแขนของหุ่นยนต์ (end-effector) ไปยังตำแหน่งเป้าหมายอย่างแม่นยำเพื่อปฏิบัติงานบนตำแหน่งนั้น ขณะเคลื่อนที่อยู่นั้นหุ่นยนต์ไม่สามารถรู้ตัวว่าได้เกิดความผิดพลาดเรื่องตำแหน่งได้ในหลายสถานการณ์ เพื่อแก้ไขปัญหานี้ เทคนิคการป้อนกลับได้ถูกนำมาใช้เพื่อปรับปรุงประสิทธิภาพการทำงานของหุ่นยนต์

การป้อนกลับแบบวงปิดเป็นวิธีการป้อนกลับที่ให้ผลดีเนื่องจากสามารถตรวจจับตำแหน่งตนเองและตำแหน่งเป้าหมายได้ การระบุตำแหน่งนี้สามารถสำเร็จได้ด้วยเทคนิคการประมวลผลภาพ หุ่นยนต์มองผ่านตาซึ่งหมายถึงกล้องนั่นเองเพื่อที่จะสำรวจตำแหน่งและทิศทาง (pose) ของตนเอง ซึ่งหมายถึงทั้งตำแหน่งของปลายแขนและตำแหน่งเป้าหมาย การระบุตำแหน่งเหล่านี้สามารถทำสำเร็จได้ด้วยการใช้เทคนิคการประมวลผลภาพหลายเทคนิคร่วมกันซึ่งเป็นส่วนที่นำเสนอในวิทยานิพนธ์นี้ด้วย เวคเตอร์เพื่อใช้แก้ไขตำแหน่งและทิศทางสามารถคำนวณได้โดยประมาณ (pose estimation) เพื่อป้อนกลับไปยังหุ่นยนต์เพื่อแก้ไข (pose correction) การเคลื่อนที่ด้วยการทำซ้ำใหม่ในครั้งต่อไป ทฤษฎี การออกแบบและการนำไปใช้งานของเทคนิคประมวลผลภาพได้บรรยายไว้ในวิทยานิพนธ์นี้

Thesis Title	A Precision Robot Arm Control for Pose Correction on PCB Using Image Processing Technique
Student	Mr. Tassanai Banlue
Student ID.	51060045
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2016
Thesis Advisor	Asst.Prof.Dr. Suthichai Noppanakeepong
Thesis Co-advisor	Assoc.Prof.Dr. Pitikhate Sooraksa

ABSTRACT

This thesis presents the design and implementation of a practical closed-looped visual servo system. The visual servo system is designed to address placement error of a robot's end-effector which is implemented in manufacturing. Real world factors in manufacturing are involved in the design and implementation. It is widely acknowledged that the placement error could be attributed to the connecting joints of a robot arms, angular errors, geometric model, and/or camera projection model. Hence, it is necessary to detect and then provide error information feedback to the robot. A common task for an automated robot is to precisely move the end-effector to a target location to perform the mission. The robot cannot recognize position error in many situations while moving the end-effector. To cope with this problem, a feedback technique must be employed in order to improve robot efficiency.

Closed-looped feedback is the most effective technique due to the detectability of own position and target position. The detection of objects of interest leads to position

identification of the own and target position. The identification can be achieved by image processing techniques which are also proposed in this thesis. Robot opens its eye (i.e., camera) to investigate the position and orientation (pose) of the own (i.e., end-effector's position) and target location (i.e., conductive test point on PCB). A pose correction vector in world coordinates can be estimated so that the feedback can help the robot correct its next movement. Brief theory, design and implementation of novel image processing techniques are also described. In addition, novel feature detection of the own position and target object are also described.

ACKNOWLEDGEMENTS

This thesis is the result of many years of work, design and implementation. Several people have contributed to the success of this work. First and foremost, I would like to express my deep gratitude towards my advisor, Asst.Prof.Dr. Suthichai Noppanakeepong. His for constant advice and support ever since I was his master degree student until now has been truly helpful. I can deeply sense his willingness for all his students to achieve success. I also would like to extend my sincerest thanks to my supervising Assoc.Prof.Dr. Pitikhate Sooraksa. His guidance and creative ideas on robotics, image processing algorithms and journal publication have been very helpful. Thanks to my bosses, Mr. Chinnawut Panyapatchoto and Mr. Panuwat Kiatdechawit who gave me the opportunity to study without interruptions to my career path of research, design and test development in industrial electronics manufacturing service (EMS) at Celestica (Thailand) Ltd. Thanks to my friends, especially Mr. Nickholas Robinson who have read this thesis and given their comments to aid the level of writing.

Thanks to my grandfather and father who were expecting my graduation but unfortunately have passed away. At least they had seen my life continually growing until their last day. I am here because of your dedication and the examples that you taught me. Thanks to my mother for showing unconditional love in taking tender care of me and her selfless actions. Thanks to my little son, Isaac, for encouraging me through being just the way you are. You are a gift from God and together with your beautiful mother have fulfilled my life. Most of all, thanks be to God for everything.

Tassanai Banlue

TABLE OF CONTENTS

Page

THAI ABSTRACT I

ENGLISH ABSTRACT III

ACKNOWLEDGEMENTS V

TABLE OF CONTENTS VI

LIST OF FIGURES IX

LIST OF TABLES XIV

CHAPTER

1 Introduction 1

1.1 Motivation and Objective 1

1.2 Uneven factors and limitation of existing technique 2

1.3 Outline of the Dissertation 3

2 Robotics for Industrial Manufacturing, Visual Servo Control 5

2.1 Introduction 6

2.2 Robotics for Industrial Manufacturing 6

2.3 Machine Vision 9

2.4 Visual Servo Control Systems 9

2.4.1 Image Based Visual Servo (IBVS) 10

2.4.2 Position Based Visual Servo (PBVS) 11

2.5 Camera Configuration 12

2.5.1 Eye-in-hand Configuration 12

2.5.2 Camera(s) Fixed in The Workspace Configuration 13
2.6 Conclusion 13

3 Robot and Image Coordinate Systems 15
3.1 Introduction 15
3.2 Camera Projection Models 16
3.3 Transformations between World 3D to Image 2D Coordinates 18
3.4 Conclusion 28

4 Machine Vision and Image Processing Techniques 30
4.1 Introduction 30
4.2 Machine Vision 30
4.3 Image Processing Techniques 31
4.3.1 Intensity Transformations and Spatial Filtering 31
4.3.2 Intensity Thresholding Techniques 34
4.3.3 Binary Morphological Techniques 39
4.3.4 Edge Detection 41
4.3.4 Pattern Matching 45
4.4 Conclusion 47

5 Feature Extraction: Probe Tip Endpoint and Test Point Identification 48
5.1 Introduction 48
5.2 Visual Closed-Loop for Solving Robotic Issues in Manufacturing 49
5.3 Probe Tip Endpoint Identification 50
5.3.1 Image Acquisition 52
5.3.2 Image Pre-Processing 52
5.3.3 Pattern Matching 54
5.3.4 Image Pose-Processing 55
5.3.5 Probe's Edge and Line Analysis 56
5.3.6 Probe's Geometry Analysis 57
5.4 Test Point Identification 61

5.4.1 Test Point Classification 65
5.4.2 Algorithm Design for Proper Via Detection 66
5.4.3 Algorithm Design for Probe Site Detection 68
5.5 Conclusion 70

6 Visual Calibration and Pose Correction 71
6.1 Introduction 71
6.2 Visual Calibration 72
6.3 Pose Estimation 78
6.4 Pose Correction 82
6.5 Conclusion 85

7 Conclusion and Remark for Future Work 86
7.1 Conclusion 86
7.2 Remark for Future Works 87

Reference 89
Appendices

Appendix A: Jacobian Iterative Estimation 93
Appendix B: Vision Software Tools 96
Appendix C: Control Point Identification 98
Appendix D: Rotational Calibration Data 102
PUBLICATIONS 106

VITA 107

LIST OF FIGURES

Figure Page

1.1 Robot accuracy issue and the steps toward pose correction of the end-effector. .2

2.1 Typical robots used in industrial manufacturing. 8

2.2 Image-based visual servo (IBVS) structure 10

2.3 Position-based visual servo (PBVS) structure 10

2.4 Relevant coordinate frames for end-effector mounted camera configuration 12

2.5 Relevant coordinate frames for end-effector fixed camera configuration 13

3.1 The geometry coordinate frame of camera projection 16

3.2 Rotation around an axis of interest follows the right hand rule 20

3.3 Illustration of transformation from one frame, {A}, to another, {B} 21

3.4 Example of a camera setup and its geometry for orthophoto 23

3.5 An acquired orthophoto of a grid object at 4000×3000 pixels resolution 23

3.6 Geometry of the object projection and camera view 24

3.7 Original perspective image taken by the camera setup 23

3.8 Orthophoto after coordinate transformation 23

3.9 Example: A 6 coordinates on an experimental model 26

3.10 Final projected points on image 28

4.1 Intensity transformations with different gamma values 32

LIST OF FIGURES (CONTINUE)

Figure Page

4.2 Image smoothing results (a) original image (b)-(c) results of smoothing with square averaging filter masks of size 3×3 and 7×7 respectively. (d) result of smoothing with a custom-mask. 33

4.3 Original gray image and binarization with fixed thresholding operation 35

4.4 Optimum Thresholding out of a gray-level histogram using Otsu's method (a) Source Image (b) Histogram of the gray image (c) $\sigma_B^2(k)$ versus gray-level (d) Output image after binarization using Otsu's thresholding value 37

4.5 Illustration of using 3×3 structuring element on a morphological function 38

4.6 Processed images using different Morphological methods 40

4.7 Result of multiple morphological steps for circles detection 41

4.8 Edge model for edge characterization 42

4.9 Example of filtering source images using Sobel masks 44

4.10 Example of intensity profile analysis 45

4.11 Geometric location of an interested object 46

4.12 Example result of pattern matching using Normalized Cross-Correlation 47

5.1 The structure of a position-based visual servo (PBVS) 48

5.2 System workflow of an implemented PBVS for a Cartesian robot 49

5.3 Pogo pin specifications 50

5.4 Image processing steps toward identifying the endpoint of the probe 51

LIST OF FIGURES (CONTINUE)

Figure Page

5.5 Images acquired with different light color (a) red light

(b) white light. 51

5.6 Result of image processing steps (a) original image cropped near interest region (b) gray image (c) pre-processed image (d) pattern matching (e)-(f) edges detection..... 53

5.7 Results of Pattern Matching with different scores, (a) Template image 220×116 pixels (b) 847.97 match (c) 855.40 match and (d) 851.69 match..... 54

5.8 Convolution filter mask for intensity highlight 55

5.9 Example computation of a filtered value using a convolution filter 56

5.10 Example result of lines and edges detection inside the best match region..... 57

5.11 (a) Geometry coordinate of cross correlation and (b) Needle pin-end characteristics. 58

5.12 Analysis of pixels intensity (a) line from start to an extended point (b) its corresponding edge strength profile 59

5.13 Pixel intensity and position of the detected edges 59

5.14 Feature classifications by the TaT algorithm 61

5.15 The take and throw algorithms for image feature extraction..... 62

5.16 Outputs in the region of interest 63

5.17 Examples of test points in gray..... 65

5.18 Ideal via with black hole extraction..... 66

LIST OF FIGURES (CONTINUE)

Figure Page

5.19 Detection outputs of test points with a black hole 68

5.20 Extraction results using multiple algorithms (a) original image (b) during image processing (c) object centers. 70

6.1 Robot coordinates system (left) and the head structure (right) 71

6.2 End-effector placement error: (a) find probe's endpoint (b) test point detection (c) select test point (d) report coordinates of probe and test point 72

6.3 Mechanical setup for rotational calibration 73

6.4 Extraction results on GUI for verification 73

6.5 (a) Calibration area; (b) possible landing locations; (c) circular model for image extraction. 74

6.6 Top view of rotational calibration 74

6.7 Sample moves of the end-effector toward the control point 77

6.8 The moving steps from q_0 toward q_f 77

6.9 Relationship of the distance on image versus the robot distance at 174 degrees 79

6.10 Rotational calibration around Z axis: (a) Linearity of rotation around Z axis (b) plot of X-Y coordinates around Z axis 80

6.11 Experimental results: (a) non-calibrated and (b) calibrated landing 80

6.12 Detection of the tip endpoint and test points at different locations 81

LIST OF FIGURES (CONTINUE)

Figure Page

6.13 Illustration of transformation of the robot's head structure.....82

6.14 Example of reference coordinates on the workspace 83

7.1 (a) Example PCB image (b) constellation of test points extracted from CAD of the example PCB 87

B.1 Vision Assistant version 8.6.0 97

B.2 Labview 8.6 included with Vision Development tools.....98

B.3 Vision Builder.....98

C.1 Center point identification out of circular four-arm model 99

C.2 Processed images for feature extraction out of circular four-arm model.....100

C.3 Sample images and identification results at different degree of calibration....101

C.4 Additional sample calibration images and identification.....102

LIST OF TABLES

Table Page

2.1 Cost and activities required for the three approaches to manufacturing 7

4.1 Most common masks of edge operators 43

6.1 Robot step sizes 75

6.2 Detection coordinates and Robot Estimation 81

CHAPTER 1

INTRODUCTION

This chapter presents an introduction to visual servoing, describing the motivation and objective of this work. The content and organization of this thesis are also surmised. Recently, visual servoing has been adopted to increase the overall accuracy of the robotic system. Visual servoing is a useful feedback loop to the robot. The system enables the robot a sense of vision on the workspace whilst the robot performs the tasks. Common steps of visual servo are as follows; camera(s) acquires an image to feed to the image processing units, the image processing units interprets meaningful features from the image, vision information from the former interpretation becomes feedback information to the robotic system. The steps are similar to the human eye(s) vision signals to the brain that is then interpreted into control data for the hand(s). Many robotics that require high imaging resolution may attach a camera to the end-effector so that the robot can visualize the target object. The setup of the configured camera to the end-effector is called "eye-in-hand configuration". The eye-in-hand configuration is adopted throughout this thesis. Basically, pixel coordinates are in two-dimensions (2D). To acknowledge a position in world coordinates (3D), it needs to transform the 2D image and it's coordinates into 3D real world coordinates prior to commanding the hand (e.g., end-effector) to move towards the target (e.g., test point). The position and orientation (pose) of the end-effector relative to the test point are identified by the proposed visual servoing system. The pose information is used to correct the position of the end-effector in the next movement. Therefore, pose correction is one major topic described in this thesis.

1.1 Motivation and Objective

This work had initially been studied for the purpose of problem solving in regards to robotic issues in manufacturing. The work was motivated to increase robot accuracy (e.g., placement error of the end-effector) as the main issue, Fig. 1a. Mostly, the scope of

the error is small and can be observed by naked eye. The accuracy issues make the robot operate with a low efficiency in servoing the end-effector to the target test point on PCB.

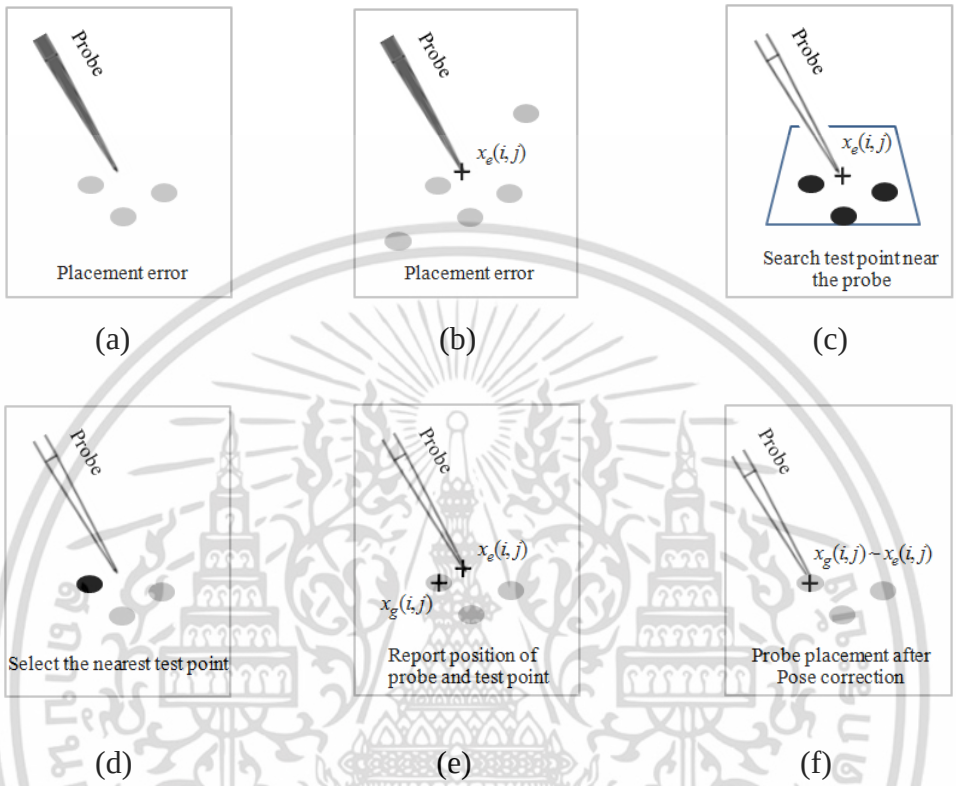


Figure 1.1 Robot accuracy issue and the steps toward pose correction of the end-effector

Fig. 1.1 shows the robot accuracy issue and the steps toward pose correction by using the proposed visual servo technique. The later chapters are intentionally described each step (i.e., Fig.1a-1f) until the placement issue is corrected. These steps in this proposed visual-servo are mainly consists of seeing and servoing. The seeing can be described in terms of image processing techniques while the servoing can be described in terms of coordinates transformation. The steps, Fig.1a-1f, are as follow:

1) The accuracy issue which is the placement error of the end-effector, Fig. 1a.
2) Find position of probe and endpoint of the end-effector (x_e), Fig.1b.
3) Search for test point positioned near the probe, Fig.1c.
4) Select the nearest test point, Fig. 1d.

5) Report position of the endpoint (x_e) and the nearest test point (x_g), Fig.1e.
6) Perform pose correction to move the end-effector to the estimated position (pose estimation).

The issues have been addressed by implementing an off-the-shelf camera together with practical solutions. Initially, behaviors of hardware are unknown and a measurement to understand all the geometric parameters of a high resolution machine is prohibitively expensive. Hence, a visual system, that is recognized, has been implemented since then. Moreover, the limits of hardware and unevenness in the manufacturing process are notoriously difficult and cannot be addressed by the hardware alone. Therefore, a vision system is added to the machine for these reasons. The work is a result of design and implementation of a visual servo using image processing techniques and machine vision in order to address both the hardware limits and the unevenness issues.

Many of the digital image processing algorithms have been studied for decades but real world applications are not yet fully included with those advanced techniques. In manufacturing, visual servo is limited to employment within the tasks that operate in well controlled environment. Complete automation of a robot employing a vision system will be reliable if it can operate with recognized target objects. Any object can be recognized by the preprogramming method (e.g., a training method). In many cases, visual servo that incorporates advanced techniques will work inconsistently under various conditions (i.e. environment changes). Certainly, the environment cannot be accurately controlled in all cases. In such cases, the unreliability of the system leads to expensive costs during operation due to maintenance, down time issues and technical services. Therefore, most of the effective reliable robotics in any industrial manufacturing will be implemented for routine tasks instead.

Currently, well calibrated machines can handle the tasks that require a high degree of precision. With repeatability a commercial robot arm and actuator can achieve an error of less than fifty micrometers. Mostly, high precision machines require calibration before use. The actuator and motor will have been tested from its manufactured origin to guarantee high accuracy. However, the errors are inevitable with the addition of components to the hardware structure and/or wiring to the motor and end-effector, which

is widely practiced. In this work, a visual servo system is proposed to work as an occasional feedback unit when the need arises instead of a full-time active unit. The feedback unit will be activated when a machine error occurs. The implemented machine is a four degree-of-freedom (4-DOF) robot. Practical design and implementation of a visual servo system is the main focus of this thesis. The design and implementation are proposed and described in later chapters.

1.2 Uneven Factors and Limitations of Existing Techniques

Every real world application has to contend with uneven factors, this fact directly affects the design, work process and software structure planning. Even though proper robot calibration improves accuracy, several parameters are often difficult to understand and are inapplicable to the robot [1]-[2]. Therefore, uneven factors in the operational process in manufacturing must be taken into account. The design and implementation has to meet the requirements of real operation. The uneven factors occur when:

1) The camera optical axis changes because of a hardware adjustment.
2) The end-effector from the supplier is not identical to the previous one, as observable from the discrepancies (in micro scale) of the acquired images.
3) The lighting conditions and glints are unpredictable since different units under test (UUT) models possess different layouts of components on the PCB.
4) The end-effector axis changes due to head structure maintenance.
5) The replacement of the existing camera either with a current model or a newer model occurs.
6) Hardware components reach the end of their lifespan.

1.3 Outline of the Doctoral Thesis

This thesis is organized into seven chapters. Most chapters contain a summary section followed by sharing experience from the practical design and implementation. In addition to this chapter, the remainder of this thesis is organized as follows:

Chapter 2 presents the literature reviews of the robotics for industrial manufacturing. Visual servo control is also introduced. This chapter covers a rather

simple introduction to both current robotics for industrial usage and visual servo. This chapter provides limited sub-topics that are relevant to the proposed visual servo control.

Chapter 3 presents coordinate transformations. This chapter focuses on the transformation of 2D image coordinates to 3D robot coordinates. Workspace is assumed to be planar and perpendicular to the X and Y axes. The robot is a single rigid body moving arbitrarily in a 3D workspace.

Chapter 4 presents literature reviews of machine vision and image processing techniques. The techniques include some traditional techniques and useful techniques which will be adopted in the proposed visual servo system. The accuracy of the pattern matching method in different environments is also discussed.

Chapter 5 presents feature extraction techniques. Implementing techniques as well as their practicality to the problem due to uneven factors in manufacturing are discussed. This chapter describes the extraction of image features, lines and center points out of the interest objects. Novel detections of probe tips and test points are described in detail. This chapter also proposes classification of test points on PCB. The center point of the test point in the 2D image coordinates are identified accurately by the proposed techniques.

Chapter 6 presents a practical design and implementation of a visual calibration. Pose estimation and pose correction are the main topics discussed in this chapter. This chapter describes the design that has been found to function well in practice in the real world environment of manufacturing. The visual servo loop does not operate full-time actively; robot will sense failure from the end-effector and then trigger the visual servo loop to literally look through a camera and re-position the end-effector with the use of visual information discovered. Complete design and implementation of an automatic calibration is also described. Chapters 5-6 have been introduced by showing constrains in current manufacturing processes and solutions but not at this stage getting into details.

Chapter 7 the conclusion of this thesis and the remarks for future work are to be discussed.

It should be noted that this work focuses on a real world design and implementation of a position-based visual servo system. Hence, the thesis refers only some background of related topics. This thesis aims to provide the explanation to the main focus, which is the practical design and implementation of the visual servo system.

CHAPTER 2

ROBOTICS FOR INDUSTRIAL MANUFACTURING AND VISUAL SERVO CONTROL

2.1 Introduction

This chapter begins by reviewing the prerequisite topics of robotics for industrial manufacturing. Machine vision and visual servo are also included in the review. Robotics has been widely employed in manufacturing and varying areas for many years. Recently, automation in manufacturing has become a major area of investment. Robotic automation is suitable for high volume manufacturing, product quality control, and increasing the speed of which new products can be brought into the market. Standard robotics and the subsequent solutions of the automated manufacturing process are continually developing. In addition, the costs of the related hardware and software are decreasing respectively. However, robots are unable to work constantly and consistently in many cases. Uneven factors occupy the initial setup which is also mentioned in this thesis. The factors that arise in robotics are the limited ability of robots to react to the changes that deviate from the initial parameters. In chapter 1, the objectives and the motivation of this thesis are also mentioned. In this chapter, the literature reviews of robotics for industrial manufacturing, machine vision and visual servo control are briefly mentioned.

2.2 Robotics for Industrial Manufacturing

The industrial robot is defined by the terms that are relevant to manipulating industrial robots operating in a manufacturing environment. The industrial robot is defined as an automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications [3]. Typical applications of robots in an industrial environment include welding, painting assembly, welding pick and place (e.g., palletizing and SMT), product inspection and automated testing. Most effective robots are preprogrammed for routine actions with predefined points in reference to position, orientation and/or a series of coordinated paths. Some robots are programmed to be more

flexible into finding the best path (e.g., shortest path) and can automatically adjust the gripper's position and orientation to correctly grasp the target object. These robots often contain machine vision in the form of their visual sensors. The vision system will be used to adjust actions and perform the intended tasks according to any useful information gained.

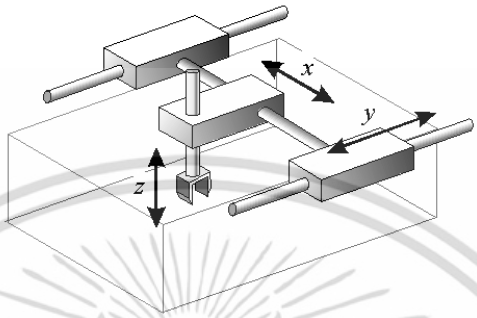
Recently, management would like to minimize operational labor costs. Robotics is frequently considered to operate in all manufacturing areas that have routine tasks, high production volumes and/or not a human friendly environment. Generally, standard (i.e., high cost) robots are the only reliable choice since they are well developed to suit the tasks at hand. They certainly come with a very high cost and are unaffordable for investment in many cases. Custom-made (i.e., low cost) robots are one very interesting solution. Most successful custom-made robots would see the robot implemented for well-defined tasks and implementing an unchanging process. One major drawback would be that the time spent in development is certainly longer than that of configuring a standard robot. However, it is valuable for the automatization of manufacturing despite the high initial costs.

Table 2.1 Cost and activities required for the three approaches to manufacturing

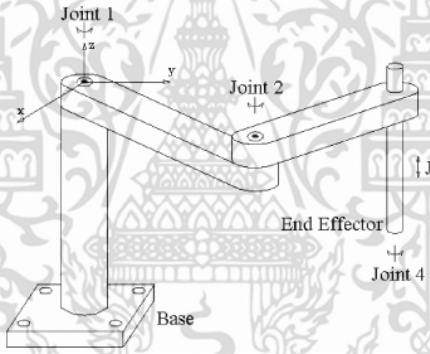
	Manual Test	Standard Robotics	Custom-made Robotics (Proposal)
Acquisition	Recruitment	Benchmark, Purchase	Design, Development
Set up	Training	Configuration	Fabrication
Sustaining mode	Skilled Labor	Maintenance	Maintenance

Table 2.1 summarizes the three different approaches [4]. In terms of component cost, the costs for custom-made robots are lower but the robot may suffer from uncertainty in positioning accuracy for because of aforementioned reasons. This is the primary reason that the visual feedback system and calibration technique is adopted to compensate for the inaccuracies in the custom-made robotics. In addition, visual servo can monitor the

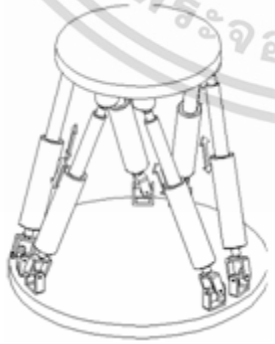
total machine error to alert the operator in order to undertake any preventive maintenance.



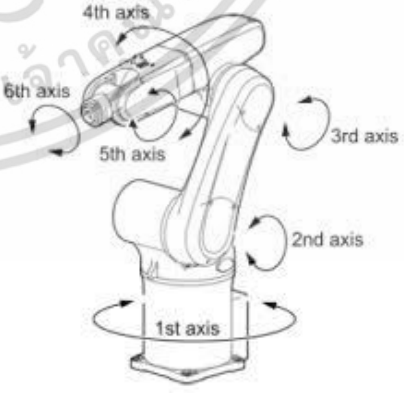
(a) Cartesian robot



(b) Scara-arm robot



(c) Parallel robot



(d) Articulated robot

Figure 2.1 Typical robots used in industrial manufacturing

Fig. 2.1(a)-2.1(d) shows typical robots which have been employed commonly in industrial manufacturing. Cartesian robot is a robot whose axes are prismatic and coincident with a cartesian coordinator, Fig. 2.1(a). Scara robot is a robot which has two parallel rotary joints travelling on a plane and another rigid actuator perpendicular to the plane, Fig. 2.1(b). Parallel robot is a robot whose arm is formed from actuators and rotary joints, Fig. 2.1(c). Articulated robot is a most common robot with at least three rotary joints, Fig. 2.1(d).

2.3 Machine Vision

Machine Vision (MV) is defined shortly as the ability "to make robots see" [5]. Machine vision (MV) provides useful outputs from the acquisition and analysis of images. The information gained must be reliable enough for robot guidance. General key steps of MV includes: 1) acquire a suitable image 2) identify the object of interest in the images 3) determine the object's position and orientation in 2D image coordinates 4) transform the 2D coordinates to robot or world coordinates 5) send the information to the robot 6) use the information to instruct the robot to move to the proper position and orientation [5].

2.4 Visual Servo Control Systems

An alternative to the increasing accuracy of robots is to adopt a visual-feedback control. Visual servoing (VS) is a technique which uses feedback information extracted from a vision sensor (i.e. camera in this thesis) to control the hand (i.e., end-effector) of a robot. In this thesis, visual servo will get feedback information of placement of the end-effector whilst a probe tip is attached to the end-effector. VS is the fusion of results from many elemental areas including image processing, kinematics, dynamics and control theory.

Visual servo control systems are mainly classified into two classes, image-based visual servo systems and position-based visual servo systems [1] and [6].

2.4.1 Image Based Visual Servo (IBVS)

An Image Based Visual Servo (IBVS) involves using information from the image to directly control the degree of freedom (DOF) [7]. Fig. 2.2 shows structure of the IBVS. The control algorithm is based on the error between current, f, and desired features, f_d, on the image plane. The algorithm does not involve any pose estimation of the target. The features may be the coordinates of visual features (e.g., corners, holes, edges and lines) or computed geometry coordinates according to the features (i.e., any structural feature that can be extracted from image) of the present object [8] and [9]. Image-based allows positioning errors to be determined, independent of robot or vision system calibration.

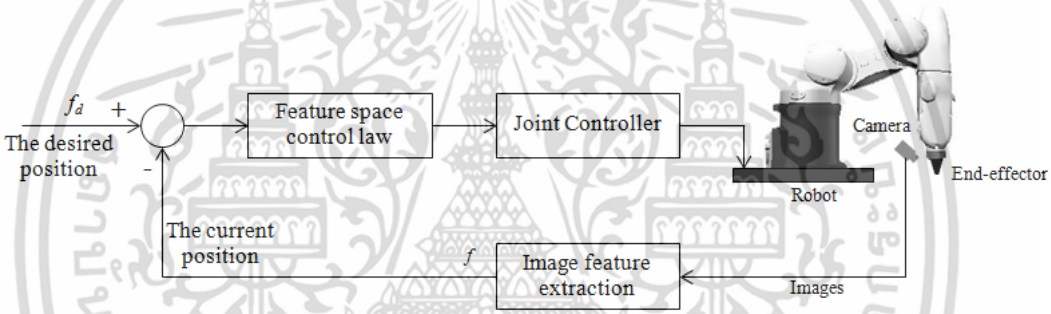


Figure 2.2 Image-based visual servo (IBVS) structure.

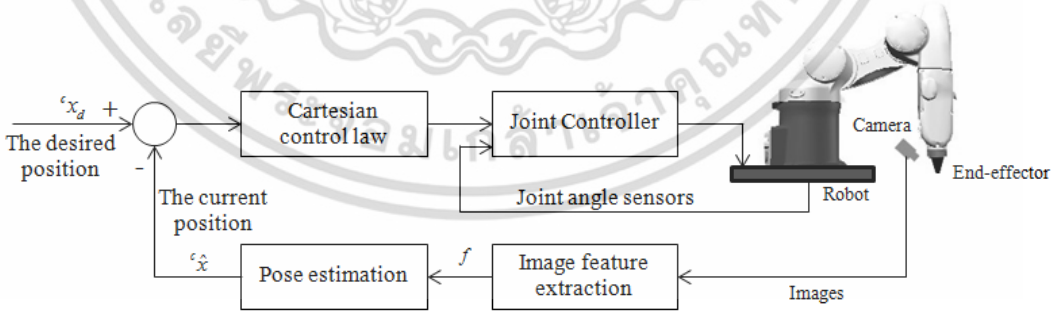


Figure 2.3 Position-based visual servo (PBVS) structure.

Although the exact geometric error cannot be absolutely determined. However, IBVS is possible to servo the robot end-effector by repeatedly executing image-based in a look-and-move fashion until convergence occurs.

2.4.2 Position Based Visual Servo (PBVS)

Position-based visual servo (PBVS) is also known as Pose-Based visual servo employing a model-based technique. Fig. 2.3 shows structure of the PBVS. The PBVS involves the geometric interpretation of the information extracted from the camera. Features of the object of interest are extracted in order to estimate the pose of the target in respect to the parameters of the camera. Absolute position is a requirement that must be known by the pose estimation before issuing a command to the robot controller. Loop of feature extraction, pose estimation and robot servo may be repeated a multitude of times until the kinematic error function is minimized. It is obviously shown that the image features are extracted as well in PBVS, but are additionally used to estimated 3D information in the Cartesian space. In contrast with Image-based, the controller still uses position feedback from the robot's joints separately from the visual feedback loop. In real application, the robot servos in close proximity to the target and then estimates the pose before servoing a final move. In comparison between the two systems, position-based control from image-based. With position based the visual feedback is computed by minimizing the kinematic errors in the estimated pose space. By comparison in image-based servoving, control values are computed on the basis of image features directly.

A typical visual servo application would be to position an end-effector relative to a target to perform a mission. In this thesis, an end-effector mounted with a camera is employed to position an end-effector to a test point (i.e., conductive pad on PCB) for injecting a test signal. Observing the end-effector directly through the camera makes it possible to sense the pose errors. Information from the visual sensing is then used for pose error correction at the next move. In general, there is no guarantee of the positioning accuracy of the system unless control points can be observed. In this thesis, both the end-effector and the target can be observed by proposed image processing techniques which will be described in a later chapter. To emphasize this distinction, the systems that only observe the target object are referred to as endpoint open-loop (EOL) systems and the

systems that observe both the target object and the robot end-effector are referred as endpoint closed-loop (ECL) systems. The ECL is emphasized and implemented in this work.

2.5 Camera Configuration

Visual servo systems typically use one of two camera configurations, (1) mounting a camera to the end-effector and (2) mount multiple cameras at fixed positions in the workspace.

2.5.1 Eye-in-hand configuration

The end-effector mounted configuration is often called an eye-in-hand configuration. This configuration has the camera mounted on the robot's end-effector. The relationship between the pose of the camera and the pose of the end-effector is constant. The relative pose can be denoted as ${}^{e}x_{c}$. The pose of the target relative to the camera frame is represented by ${}^{c}x_{t}$. The relationship between these poses is shown in Fig 2.4. Relevant coordinate frames are consisting of world, end-effector, camera and target coordinates.

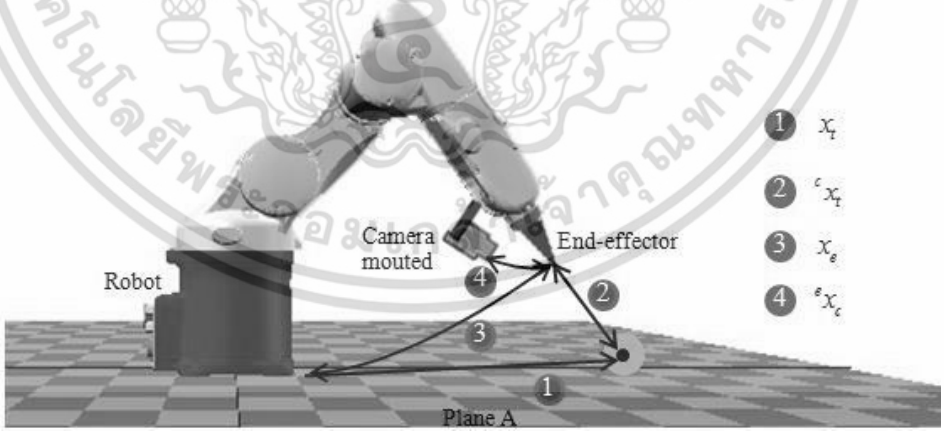


Figure 2.4 Relevant coordinate frames for end-effector mounted camera configuration

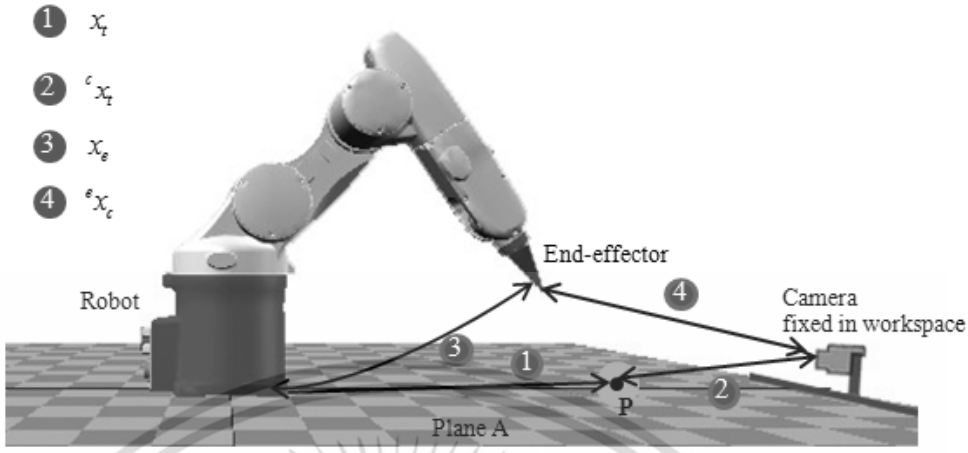


Figure 2.5 Relevant coordinate frames for end-effector fixed camera configuration

2.5.2 Camera(s) Fixed in The Workspace Configuration

In this configuration, the camera(s) are fixed in the workspace. The camera(s) are related to the base coordinate system of the body by ${}^0 x_c$ and to the object by ${}^c x_t$. The relationship between these poses is shown in Fig 2.5. The camera image of the target is independent from the motion of the robot. A camera is fixed to the body is most basic example case. Hence, extrinsic parameters are constant. The parameters can be established by using a camera calibration procedure. The end-effector may obstruct camera view in some cases. To increase visibility of the camera, the camera can be fixed to another movable part. Multiple cameras should also be considered for best view selection. In addition, camera pan, tilt and zoom can be included in order to observe the target object with the best possible view.

2.6 Conclusion

Certainly, custom-made robotics has it place in manufacturing. Custom robotics may be suitable in many initial phases when moving from labor-based manufacturing to automatized manufacturing. Specific routine tasks can be considered for replacement with custom robotics. Investment in Research and Development (R&D) must be taken into account. A high volume product with routine manufacturing tasks should be the first

area to explore for the possibility of implementing a custom-made robot. Currently, robotic components, actuator, high accuracy motors, software driver interface and programming tools are ready like never before for development. The development engineer can consider configuring the robotic components to serve specific requirements of automated manufacturing. A simple Cartesian robot is a good start for many applications also.

As result, management has been desired to move towards automated manufacturing lately. Hence, Cartesian's robots have been implemented for a number of routine tasks. Low accuracy like automatic insertion is the very first opportunity of simple robotics. Later on, higher precision robots are required to work in micro-scale. Finally, visual servo is required and implemented for pose correction to the placement error of the robot's end-effector.

In this work, absolute position is required in order to perform pose correction. Also, visual system requires close look to the target object (i.e., test point). Therefore, the position-based visual servo with the eye-in-hand configuration are considered for this work.

CHAPTER 3
ROBOT AND IMAGE COORDINATE SYSTEMS

3.1 Introduction

Visual servo system is mainly consists of the seeing and the servoing. To servo (i.e., move) robot end-effector, the hardware controller requires an absolute position. Often, reference positions are in different coordinates. Hence, coordinate systems and their transformations are described in this chapter. A visual sensor (e.g., camera) detects the object of interest which is located in a workspace. Information about the position of the object is extracted from its image features. The interest position on image (i.e., pixel coordinates) is still in image coordinates. Coordinate transformation is used to transform the image coordinates to robot coordinates to enable further control. Hence, transformation between image coordinates and robot coordinates is important and is therefore emphasized in this chapter. The task space of the robot is represented by T. The task space is a set of positions and orientations. Typically, a robot would be a single rigid body moving in a three-dimensional workspace. The task space that is restricted to a subspace of SE^3 can be written as [1]:

$$T = SE^3 = \Re^3 \times SO^3 \tag{3.1}$$

where $\Re^3$ is a pure translation and SO^3 represents the rotations containing the coordinates of each call to the feedback loop.

Typically, visual sensors supply information of the location of an object of interest with respect to a camera frame, while the configuration used to locate the object is specified with respect to a coordinate frame attached to the object. Notation of a translation from original frame $\{y\}$ to coordinate frame $\{x\}$ in general form can be written as:

$${}^{x}P = {}^{x}x_{y}\ {}^{y}P = {}^{x}R_{y}\ {}^{y}P + {}^{x}t_{y} \tag{3.2}$$

where an original point P with respect to coordinate frame x and y is denoted by ${}^{x}P$ and ${}^{y}P$. The rotation matrix that represents the orientation of frame y with respect to frame x is

denoted by ${}^{x}R_{y}$. ${}^{x}t_{y}$ represents the location of frame y to frame x. ${}^{x}x_{y}$ represents both the position and the orientation from the original frame to another frame.

3.2 Camera Projection Models

This sub-section provides understanding of geometric aspects of the imaging process. The geometry and mathematics of the pinhole camera can explain projection from an object in 3D world coordinates into 2D image coordinates, see Fig. 3.1.

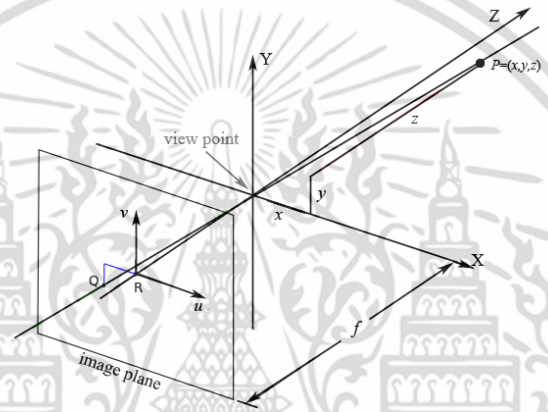


Figure 3.1 The geometry coordinate frame of camera projection

The pinhole camera model describes the mathematical relationship between the coordinates of a 3D point, ${}^{x}P=P(x,y,z)$, and its projection onto the camera coordinate frame, c. Therefore, coordinate transformation of the 3D point to 2D point, ${}^{c}P$, can be denoted as:

$$ {}^{c}P = {}^{c}x_{x}\,{}^{x}P \tag{3.3} $$

and

$$ {}^{c}P = {}^{c}Q(u,v) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \tag{3.4} $$

while $u=fX/Z$ and $v=fX/Z$. Fig. 3.1 shows geometry of projection of an object at ${}^{x}P(x,y,z)$ to image plane ${}^{c}Q(u,v)$ while Z is referred to as the optical axis. An image plane is the plane parallel to axes X and Y, where the 3D object is projected through the aperture of the camera. The image plane is located at a distance of f behind the center of projection, O. f is referred to as the focal length of the pinhole camera. World, P, coordinates can be re-written in matrix multiplication form as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3.4}$$

Assuming that the 2D point is in homogeneous coordinates, divide through by the last element

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \; x = x_1 / x_3, \; y = x_2 / x_3 \tag{3.5}$$

To produce a complete perspective projection, camera intrinsic and camera extrinsic parameters must be considered. All the intrinsic parameters can be written in a matrix K as:

$$K = \begin{pmatrix} f / s_x & 0 & c_x \\ 0 & f / s_y & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.6}$$

where s_x and s_y is the size of a pixel in mm. Location c_x and c_y is optical center of the image. Alternatively, f_x and f_y in pixels can be used. Eq. 3.5 can be rewritten as:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = K \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} {}^{C} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{3.7}$$

3.3 Transformations between World 3D to Image 2D Coordinates

To transform 3D points in world coordinates, ${}^{W}P$, to camera coordinates, ${}^{C}P$, the transformation can be expressed as:

$$ {}^{C}P = {}^{C}H_{W}\,{}^{W}P = \begin{pmatrix} {}^{C}R_{W} & {}^{C}t_{W} \\ 0 & 1 \end{pmatrix} {}^{W}P, \tag{3.8} $$

where H represents a single transformation homogenous matrix which consists of rotation R and translation t. The terms rotation, R, and translation, t, can be written in the form of the extrinsic camera matrix, M_{ext}, as:

$$ M_{ext} = \begin{pmatrix} {}^{C}R_{W} & {}^{C}t_{w} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{pmatrix}, \tag{3.9} $$

where r_{11}, r_{12}, r_{13},…, r_{33} are elements of general rotation matrix. Hence, Eq. 3.7 can be rewritten to complete perspective projection. Now, a 3D world coordinate (i.e., ${}^{W}P$) is projected to a point in pixel image (x_{im}, y_{im}).

$$ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = KM_{ext} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}^{W}, \tag{3.10} $$

where $x_{im}=x_1/\ x_3$ and $y_{im}=x_2/\ x_3$. The pose of frame $\{B\}$ with respect to frame $\{A\}$ is described by rotation vector R and translation vector t. When R is rotation 3×3 matrix that represents 3 angles. The matrix can be obtained from these three elemental rotations using matrix multiplication.

$$ R = R_{\theta_z} R_{\theta_y} R_{\theta_x} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{3.11} $$

The first rotation is the rotation about the X axis, from frame {A} to {B}.

$$\begin{pmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & -\sin\theta_X \\ 0 & \cos\theta_X & \cos\theta_X \end{pmatrix} \begin{pmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{pmatrix}; \; R_{\theta_X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & -\sin\theta_X \\ 0 & \cos\theta_X & \cos\theta_X \end{pmatrix}, \tag{3.12}$$

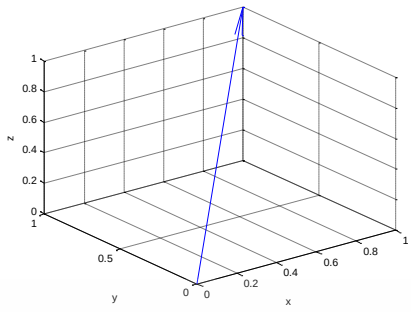
Rotation about the Y axis from frame {A} to {B} can be written as

$$\begin{pmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{pmatrix} = \begin{pmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{pmatrix} \begin{pmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{pmatrix}; \; R_{\theta_Y} = \begin{pmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{pmatrix}, \tag{3.13}$$

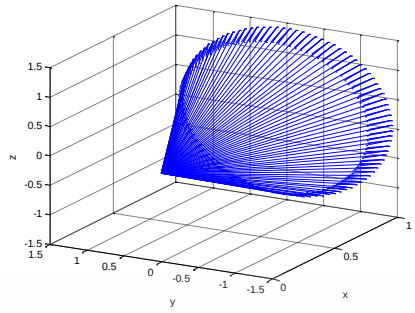
Rotation about the Z axis from frame {A} to {B} can be written as

$$\begin{pmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{pmatrix} = \begin{pmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{pmatrix}; \; R_{\theta_Z} = \begin{pmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.14}$$

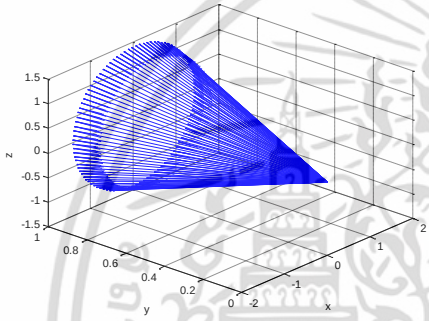
The direction of rotation, the rotation of a unit vector about an axis follows the right hand rule. Fig. 3.2 illustrates the rotation of a unit vector, Fig. 3.2a, around each axis, Fig3.2b-Fig.3.2d. The three rotations matrix can be concatenated to yield a single 3×3 general rotation matrix, $R=R_ZR_YR_X$.



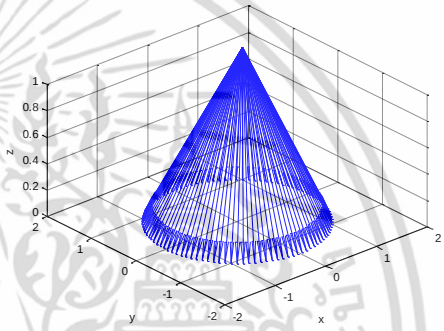
(a) A unit vector



(b) unit vector rotated about the X axis



(c) Unit vector rotated about the Y axis



(d) unit vector rotated about the Z axis

Figure 3.2 Rotation around an axis of interest follows the right hand rule

Rotational transformation of frame {A} to frame {B} can be represented by

$$ {}^{B}R_{A} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, \tag{3.14} $$

A vector in frame {A} can be rotated to obtain its representation in frame {B}

$$ {}^{B}v = {}^{B}R_{A}\,{}^{A}v, \tag{3.15} $$

As in the rotation in 2D coordinates, rotation matrices are orthonormal so the inverse of a rotation is just its transpose

$$\left({}^{B}R_{A}\right)^{-1} = {}^{B}R_{A}\,{}^{A}v, \tag{3.16}$$

The elements of R are the projections of unit vectors from one frame onto the unit vectors of the other frame. So the columns of R are the unit vectors of A, expressed in the B frame

$$\left({}^{B}R_{A}\right) = \left(\left({}^{B}\hat{x}_{A}\right)\left({}^{B}\hat{y}_{A}\right)\left({}^{B}\hat{z}_{A}\right)\right) = \begin{pmatrix} \left({}^{A}\hat{x}_{B}{}^{T}\right) \\ \left({}^{A}\hat{y}_{B}{}^{T}\right) \\ \left({}^{A}\hat{x}_{B}{}^{T}\right) \end{pmatrix}, \tag{3.17}$$

Similar to Eq. 3.2, transformation of a point ${}^{B}P$ from coordinate frame {B} to frame {A} can be expressed in terms of R, t as

$${}^{A}P = {}^{A}x_{B}\,{}^{B}P = {}^{A}R_{B}\,{}^{B}P + {}^{A}t_{B} \tag{3.18}$$

The rotation followed by a translation can be written in rigid transformation as

$${}^{x}P = {}^{x}x_{y}\,{}^{y}P = {}^{x}R_{y}\,{}^{y}P + {}^{x}t_{y} \tag{3.19}$$

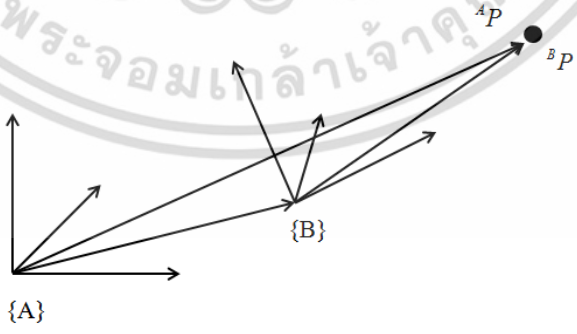


Figure 3.3 Illustration of transformation from one frame, {A}, to another, {B}

Fig. 3.3 depicts a rotation followed by translation from one frame to another. The rigid transformation matrix can be represented by a single 4×4 homogeneous transformation matrix as

$$ {}^{B}R_{A} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x_0 \\ r_{21} & r_{22} & r_{23} & y_0 \\ r_{31} & r_{32} & r_{33} & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.20} $$

Therefore

$$ {}^{A}P = {}^{A}x_{B}\,{}^{B}P = {}^{A}R_{B}\,{}^{B}P + {}^{A}t_{B} = {}^{A}H_{B}\,{}^{B}P \tag{3.21} $$

The matrix inverse is the inverse transformation

$$ {}^{A}H_{B} = \left({}^{B}H_{A}\right)^{-1} \tag{3.22} $$

The transformation can be written in the form, $Ax=b$, where

$$ A = \begin{pmatrix} x_A^{(1)} & -y_A^{(1)} & 1 & 0 \\ y_A^{(1)} & x_A^{(1)} & 0 & 1 \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ y_A^{(N)} & x_A^{(N)} & 0 & 1 \end{pmatrix}, \; x = \begin{pmatrix} c \\ s \\ t_x \\ t_y \end{pmatrix}, \; b = \begin{pmatrix} x_B^{(1)} \\ y_B^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ y_B^{(N)} \end{pmatrix}, \tag{3.23} $$

Typically, a camera of the eye-in-hand system is not configured to be orthogonal to the target object plane. Fig. 3.4 show the camera setup which is orthogonal to a target object shown in Fig. 3.5. However, orthophoto can be regenerated using a 2D rigid transformation. Assuming no lens distortion, measurement of the distance of the object of interest on orthophoto can be considered in linear scale. Hence, orthophoto regeneration is a method employed for many applications.

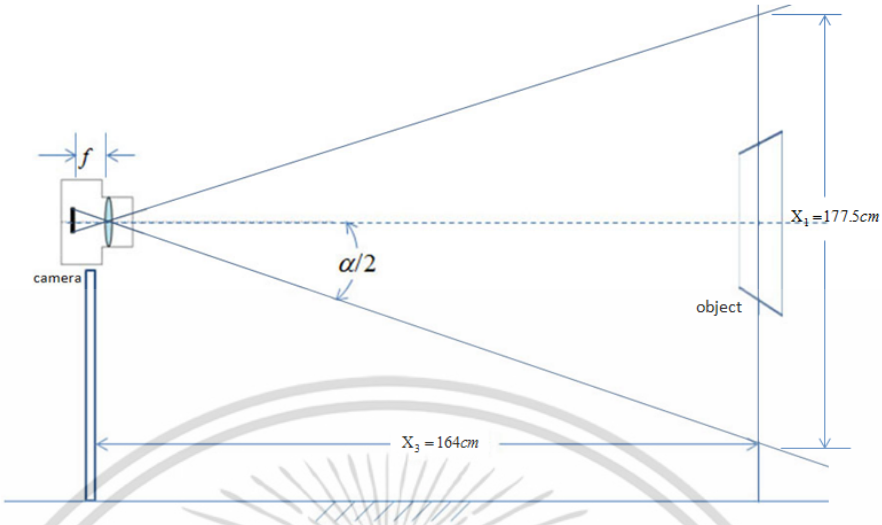


Figure 3.4 Example of a camera setup and its geometry for orthophoto

Figure 3.5 An acquired orthophoto of a grid object at 4000×3000 pixels resolution

From basic geometry of the above camera (i.e., Finepix S200EXR) setup, The angle of view can be calculated from $\beta = 2\tan^{-1}\left(\left(T_w / Z\right)\left(V_u / U_w\right)\right)$ when T_w is the width in *mm* of the target object (i.e., grid paper's width), Z is the distance between the camera and the

target object, V_u is the image width in pixels and U_w is the object size in pixels. Roughly, the horizontal frame of view, β, is about 56.26 degrees. In the same way, the angle of the vertical frame of view, α, will be 43.4 degrees.

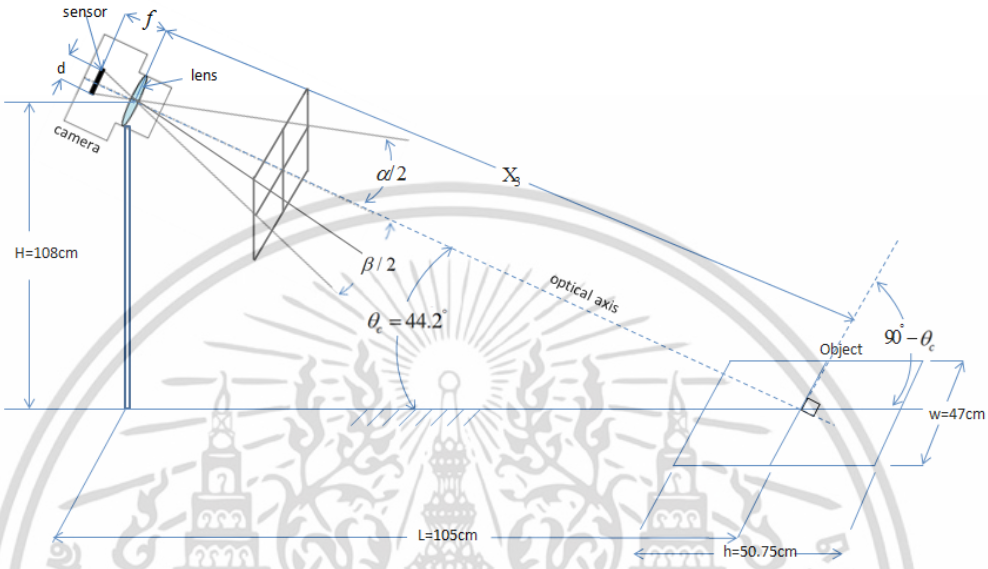


Figure 3.6 Geometry of the object projection and camera view

The camera geometry as seen in the above picture (Fig. 3.6) mimics a typical setup of the eye-in-hand configuration. Mostly the optical axis of the eye-in-hand configuration is not perpendicular to the object plane. Therefore, the acquired image will become perspective (non-orthophoto) on image coordinates. Fig. 3.7 shows example image acquired by the camera setup in Fig. 3.6. Using basic trigonometry, the camera angle can be written as:

$$\alpha = 2\tan^{-1}\frac{d}{f}$$

where d represents size of the camera sensor in the measured direction. For example most of the sensors are a rectangular shape, the size in pixels is horizontal, d_x, and vertical , d_y, can be used to obtain the horizontal and vertical angles.

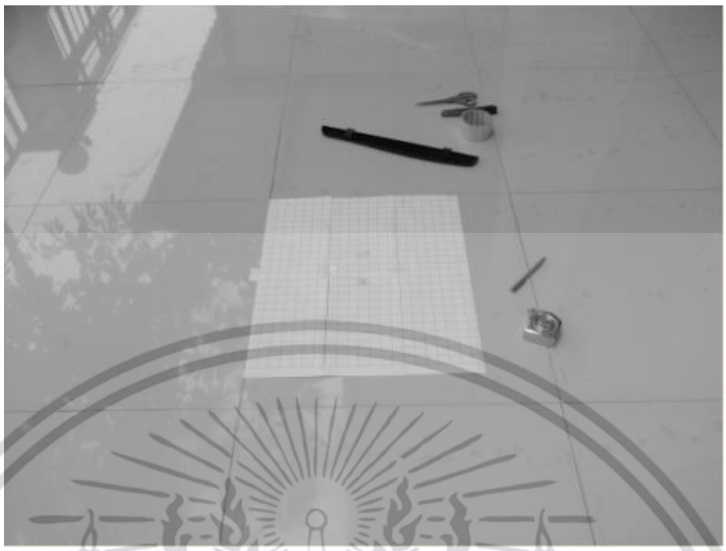


Figure 3.7 Original perspective image taken by the camera setup

Figure 3.8 Orthophoto after coordinate transformation

Fig. 3.8 shows the orthophoto after regeneration by using coordinate transformation. In Eq. 3.8 world coordinates, ${}^{W}P$, can be replaced with model coordinates, ${}^{M}P$, and become

$$\tilde{p} = KM_{ext}\,{}^{M}P = K\left({}^{C}R_{W} \quad {}^{C}t_{W}\right){}^{M}P \tag{3.24}$$

where ${}^{M}P$, is a set of coordinates in 3D of a target known model, and $\tilde{p}$ is a set of coordinates on image.

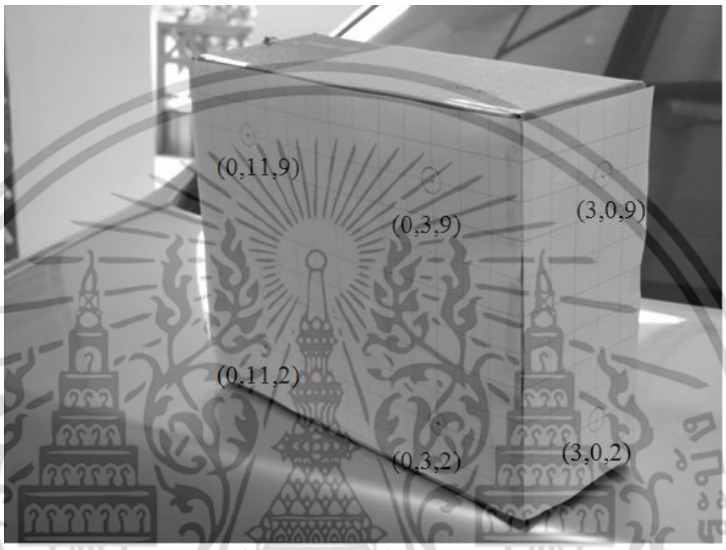


Figure 3.9 Example: A 6 coordinates on an experimental model

The set points on the model in Fig. 3.9 can be constructed as

$${}^{M}P = \begin{pmatrix} X_1 & X_2 & \dots & X_6 \\ Y_1 & Y_2 & \dots & Y_6 \\ Z_1 & Z_2 & \dots & Z_6 \\ 1 & 1 & \dots & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 3 & 0 & 0 & 3 \\ 11 & 3 & 0 & 11 & 3 & 0 \\ 9 & 9 & 9 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{3.25}$$

where c_x and c_y are 2000 and 1500 respectively, found at the center location of the image 4000×3000 pixel resolution. From the geometry model of the camera view, Fig. 3.6, the focal length f_x and f_y can be written as:

$$f_x = \frac{c_x}{\tan\left(\left(\beta/2\right)\left(\pi/180\right)\right)} = \frac{2000}{\tan\left(\left(56.2/2\right)\left(\pi/180\right)\right)} \cong 5026, \tag{3.26}$$

and

$$f_y = \frac{c_y}{\tan\left(\left(\alpha/2\right)\left(\pi/180\right)\right)} = \frac{1500}{\tan\left(\left(43.4/2\right)\left(\pi/180\right)\right)} \cong 2806, \tag{3.27}$$

Then, the intrinsic matrix K can be constructed as

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 5026 & 0 & 2000 \\ 0 & 2806 & 1500 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.28}$$

Then, construct extrinsic, M_{ext}, matrix using Eq. 3.11 throughout Eq. 3.14.

$$M_{ext} = \begin{pmatrix} {}^{C}R_{W} & {}^{C}t_{w} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{pmatrix} = \begin{pmatrix} 0.7403 & -0.6702 & -0.0535 & 4.2125 \\ -0.2788 & -0.2336 & -0.9315 & 6.7617 \\ 0.6118 & 0.7045 & -0.3598 & 23.6748 \end{pmatrix}, \tag{3.29}$$

where $\theta_x, \theta_y, \theta_z$, t_x, t_y and t_z are 2.04,-0.65,-0.36,4.21,6.76 and 23.67 respectively. They are obtained through an estimation method (i.e., the Jacobian iterative estimation) which will be described in an appendix section. Finally, projected points, $\tilde{p} = KM_{ext}\,{}^{M}P = K\begin{pmatrix} {}^{C}R_{W} & {}^{C}t_{W} \end{pmatrix}{}^{M}P$, from model onto the image will be

$$\tilde{p} = \begin{pmatrix} 5026 & 0 & 2000 \\ 0 & 2806 & 1500 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.7403 & -0.6702 & -0.0535 & 4.2125 \\ -0.2788 & -0.2336 & -0.9315 & 6.7617 \\ 0.6118 & 0.7045 & -0.3598 & 23.6748 \end{pmatrix} \begin{pmatrix} 0 & 0 & 3 & 0 & 0 & 3 \\ 11 & 3 & 0 & 11 & 3 & 0 \\ 9 & 9 & 9 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 38073 & 53747 & 74457 & 44992 & 60666 & 81375 \\ 21212 & 22151 & 21053 & 57760 & 58700 & 57601 \\ 28.18 & 22.54 & 22.27 & 30.70 & 25.06 & 24.79 \end{pmatrix} \quad (3.30)$$

where $x_{im}=x_1/\ x_3$ and $y_{im}=x_2/\ x_3$. Therefore, the final projected point on image of the first point, by way of this example, is (38073/28.18,21212/28.18)=(1360,753). Fig. 3.10 shows the six projected points on the image model.

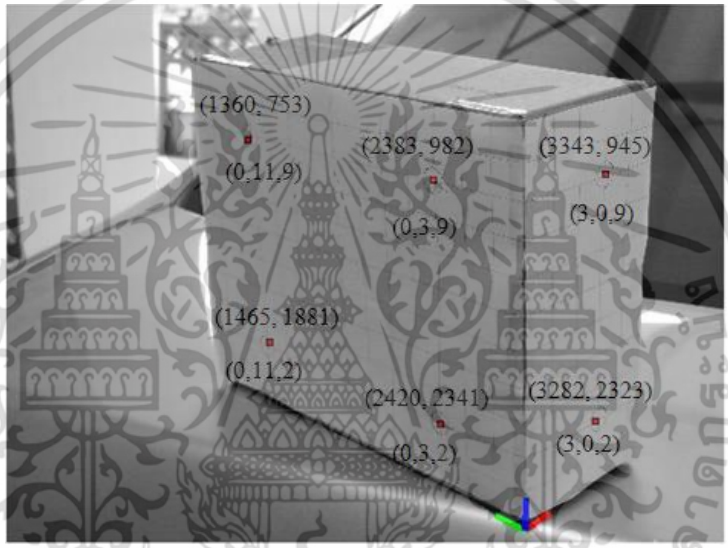


Figure 3.10 Final projected points on image

3.4 Conclusion

Complete transformations between 3D to 2D coordinates can be achievable when all geometric parameters are known. Most of transformations described in this chapter can be used with the proposed work. In practice, it is unnecessary to know all of the geometry parameters (i.e., intrinsic and extrinsic parameters). Actually, measurement to read all of the geometry parameters is not particularly practical and often expensive. In addition, the parameters would change to unknown anyway after any adjustment of the hardware during setup and/or maintenance. Hence, the fixed parameters will be effectively utilized for the transformations of the hardware which has no change to its

structure. In this work, most of the transformations were applicable at the beginning after finishing all the hardware setup only. Therefore, the parameters which has chance to change (e.g., camera parameters) will require replacement with a more practical method. The method is required for solutions to this problem in regards to the real world applications of robotics. The practical method that is required needs proper planning and support from both hardware design and software design.

CHAPTER 4
MACHINE VISION AND IMAGE PROCESSING TECHNIQUES

4.1 Introduction

This chapter describes machine vision towards a number of image processing techniques which are essential to the seeing of the proposed position-based visual servo system (PBVS). The main purposes of machine vision are automatic inspection and machine guidance. In the proposed PBVS, machine vision recognizes target objects on a two-dimensional image and then estimates error distance and orientation. Then, pose correction can be made later by using the relative position of the current robot and the position of the target object. A machine vision system is an application used in image processing to create a model of the real world from images. Most image processing techniques involve the analysis of the intensities or gray levels of pixels with respect to their position in the image, there are many techniques which may be appropriate for different tasks. This chapter reviews a number of classical techniques.

4.2Machine Vision

A machine vision system recovers useful information about a scene from its two-dimensional projections. The recovered information is not directly available and must be mapped into a three-dimensional world by using knowledge of the objects in the scene and projection geometry. Machine vision is widely practiced in various applications. The primary uses for machine vision in industrial applications are for robot guidance and automatic inspection. Usually, *image processing* would transform images into other images. *Machine vision* employs image processing techniques to extract the required information and then make decisions such as which objects to select as the target object under tracking including relative geometry information between the robot and the target object [12].

4.3 Image Processing Techniques

This section briefly reviews image processing techniques that are fundamental to a typical machine vision application(s) [12-16]. Most of the reviewed techniques are integrated into the proposed visual system.

4.3.1 Intensity Transformations and Spatial Filtering

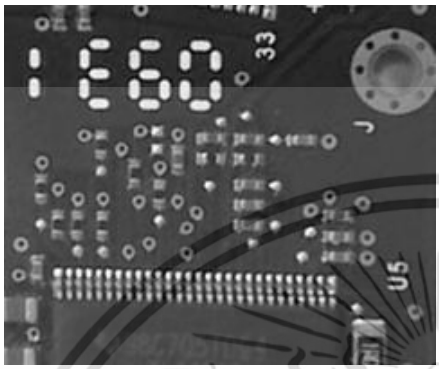
The term spatial domain refers to the image plane itself. It relies on the analysis of pixels according to their position in the image. Two main categories of spatial domain processing are intensity transformations and spatial filtering. The processes are denoted as.

$$g(x,y) = T[f(x,y)] \tag{4.1}$$

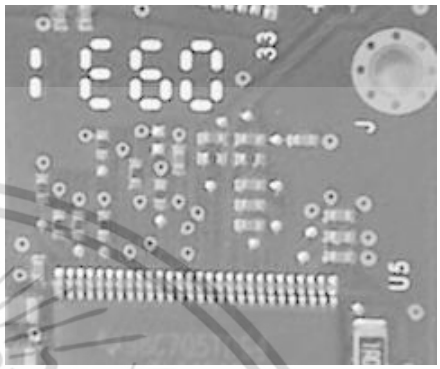
where $f(x,y)$ is the source image, $g(x,y)$ is the processed image. T is an operator on f at corresponding location (x,y) in the images. Gamma correction is an example of a point operation. The gamma specifies the shape of the curve that maps the intensity values in f to create g. If the gamma is less than 1, the mapping is weighted towards higher output values to enhance the contrast of high intensities. If gamma is greater than 1, the mapping is weighted towards lower output values to enhance the contrast of low intensities. $s=f(r)$ represents point operations, it maps input pixel value r to output value s. Fig. 4.1 shows output of image after applying with different gamma value. In practice, gamma may be adjusted within a range to find if edge and line can be promoted in the acquired image.

Instead of transforming a single pixel with an intensity value to a new value, spatial filtering considers the value of neighborhood. It is termed neighborhood processing or spatial convolution. The process consists of moving the center of the mask w with size $m \times n$ from point to point in an image, f, with image size $M \times N$, while the mask and image indexing start from 0. At each point (x,y), the sum of products of the mask coefficients with corresponding pixels under the mask, the mask is then slid over the image and operation of the sum of products is repeated to produce an output image $g(x,y)$. This is also called a cross-correlation where the mask is flipped both horizontally and vertically.

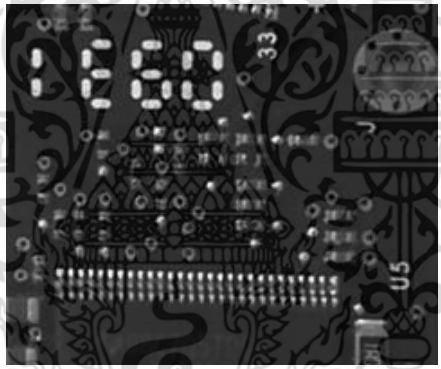
$$g(x,y) = \sum_{s=-m/2}^{m/2} \sum_{t=-n/2}^{n/2} w(s,t) f(x+s, y+t), \tag{4.2}$$



(a) Original image, gamma=1



(b) Output image using gamma<1 (0.36)



(c) Output image using gamma>1 (1.6)

Figure 4.1 Intensity transformations with different gamma values

In spatial filtering, convolution with averaging filter is most commonly used for reducing noises. However, expanding the size of the filter mask will result in blurring of the image, Fig. 4.2(b)-(c). The mask can be customized according to the objective of the filtering operation. Instead of smoothing pixels, a pixel can be highlighted by giving the central element of the mask a larger value than the sum of all its neighbors, Fig. 4.2(d). The mask is [-1 -1 -1;-1 10 -1;-1 -1 -1], for this example. This highlighting technique is adopted to sharpen the edges of an object in gray scale image before finding their edges.

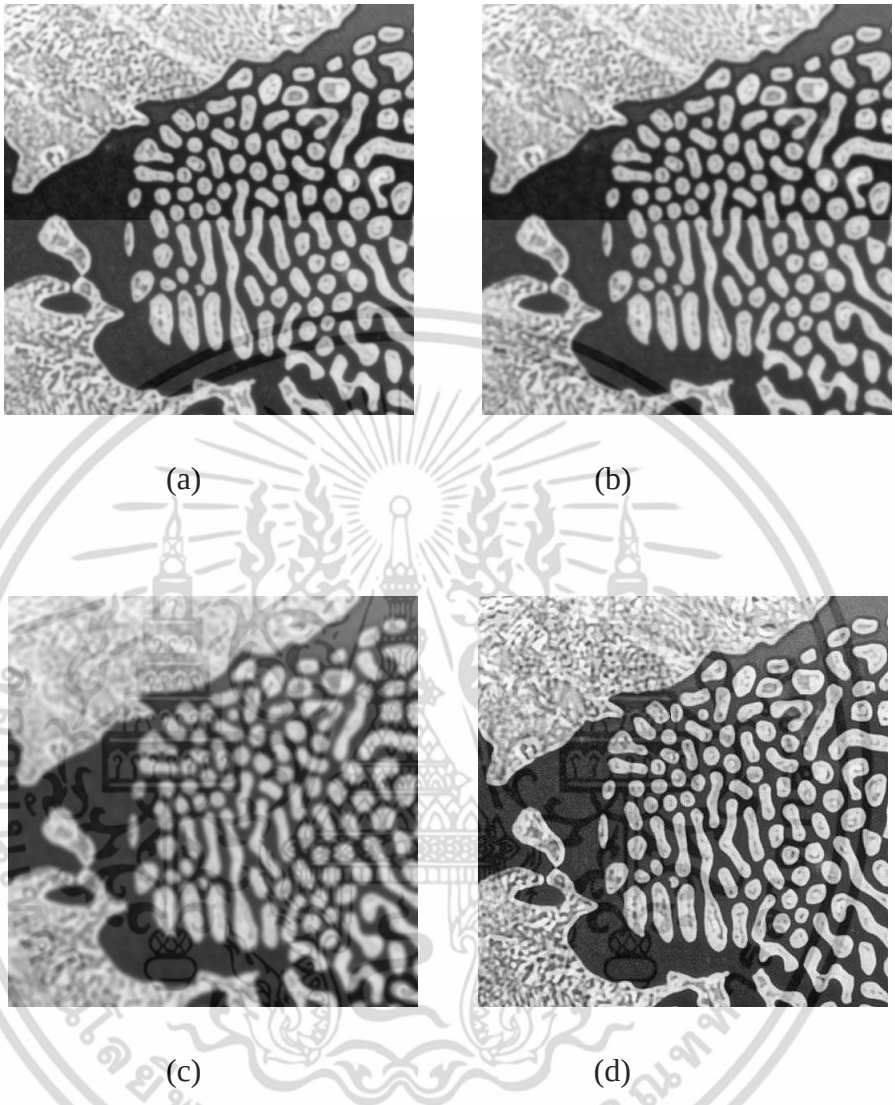


Figure 4.2 Image smoothing results (a) The original image (b)-(c) The results of smoothing with square averaging filter masks of sizes 3×3 and 7×7 respectively. (d) The result of smoothing with a custom-mask.

4.3.2 Intensity Thresholding Techniques

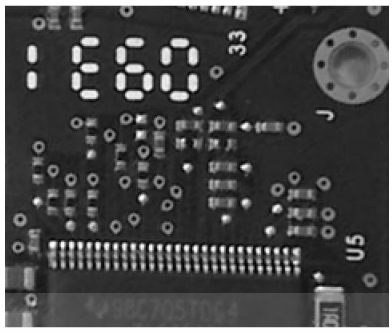
Intensity thresholding can be used to convert a grayscale image to binary images. The thresholding is a most straightforward image processing operation. It is the simplest method of image segmentation. Generally, the segmentation is to separate objects from their background. Thresholding is defined by two parameters; *Lower Threshold* and *Upper Threshold*. All pixels that have gray-level values equal to or greater than the *Lower Threshold* and equal to or smaller than the *Upper Threshold* are assigned as pixels belonging to particles [12],[15].

4.3.2.1 Manual or Fixed Thresholding

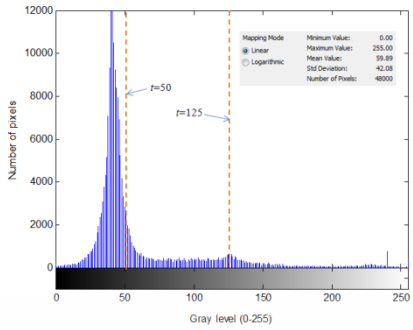
The operation converts an image I_o with a predetermined threshold t to a new image I_t and can be denoted for an image size $X \times Y$ as:

$$I_t(x,y) = \begin{cases} 1, & I_O(x,y) \geq t \\ 0, & I_O(x,y) < t \end{cases} \quad x \in \{0,1,...,X-1\}, y \in \{0,1,...,Y-1\} \tag{4.3}$$

Fig. 4.3(a)-4.3(b) shows a cropped PCB in grayscale and its histogram respectively. Fig. 4.3(c)-4.3(d) shows binary images which are the result of the thresholding iteration with thresholding parameters of [50, 255] and [125, 255] respectively. The image histogram shows clearly regions of brighter particles which mostly are conductive copper (i.e., test point and lead) and also the white labels which are of the same size as the test points are. The binary image of lower t value will preserve information of target particles but the particle area will potentially be connected to a neighbor (Fig. 4.3(c)) while a higher t value can perform better object segmentation but it causes lost particle information under interest (Fig. 4.3(d)).



(a) Original gray image



(b) Histrogram of the gray image

(c) Binary image with t=50

(d) Binary image with t=125

Figure 4.3 Original gray image and binarization with a fixed thresholding operation

4.3.2.2 Automatic Thresholding

In contrast to manual thresholding, automatic methods do not require the setting of the lower and upper values. Instead, the automatic techniques are adaptive or dynamic. They are well suited for conditions in which the light intensity varies. The varying intensity causes problems in segmenting an image into particle and background regions, the boundaries are not easily distinguished. The algorithms for thresholding to obtain binary images can be generalized to more than two levels. Automatic thresholding is more suitable in different applications. In thought cases, images require enhancement by preprocessing steps first before thresholding. Then, morphogy techniques can help to separate particles in the binary image after binarization by the thresholding.

Clustering is the most frequently used and general-purpose threshold method. Even though, certain algorithm hard-coded in NI clustering is not revealed. Among clustering methods, the Otsu method is one of the most referenced [17],[22]. Thresholding values calculated by NI's and Otsu's method are mostly the same. Hence, Otsu clustering is highlighted for further explanation.

Let L represent gray level [0,1,2,…,L-1] when L is 256 for 8 bit grayscale. n_i and N is the number of pixels at level i and total number of pixels respectively. The histogram of the gray-level can be normalized as probability distribution as:

$$p_i = \frac{n_i}{N},\ p_i \geq 0,\ \sum_{i=1}^{L} p_i = 1 \tag{4.4}$$

$$\omega(k) = \sum_{i=1}^{k} p_i \tag{4.5}$$

$$\mu(k) = \sum_{i=1}^{k} i p_i \tag{4.6}$$

and

$$\mu_T = \mu(L) = \sum_{i=1}^{L} i p_i \tag{4.7}$$

$$\sigma_B^2(k) = \frac{\left(\mu_T \omega(k) - \mu(k)\right)^2}{\omega(k)\left(1-\omega(k)\right)} \tag{4.8}$$

and subsequently the final optimal threshold level is

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \tag{4.9}$$

The threshold level can be scaled to the range [0, 1] as

$$T_{otsu} = \frac{\sigma_B^2(k^*) - 1}{L - 1} \tag{4.10}$$

Fig. 4.4(a) is an example histogram of a gray image with size 67×67 pixels. Therefore, total pixels, N, are 4489, while n_i are in form of histogram of the image intensity. Then, p_i is the histogram by the N=4489 constructing an array of L elements, where $\omega(k)$ and $\mu(k)$ represents zeroth and the first-order cumulative moments of p_i up to kth level respectively. μ_T is the last element of the cumulative sum of p_i multiplied by i [1,2,3,…,L].

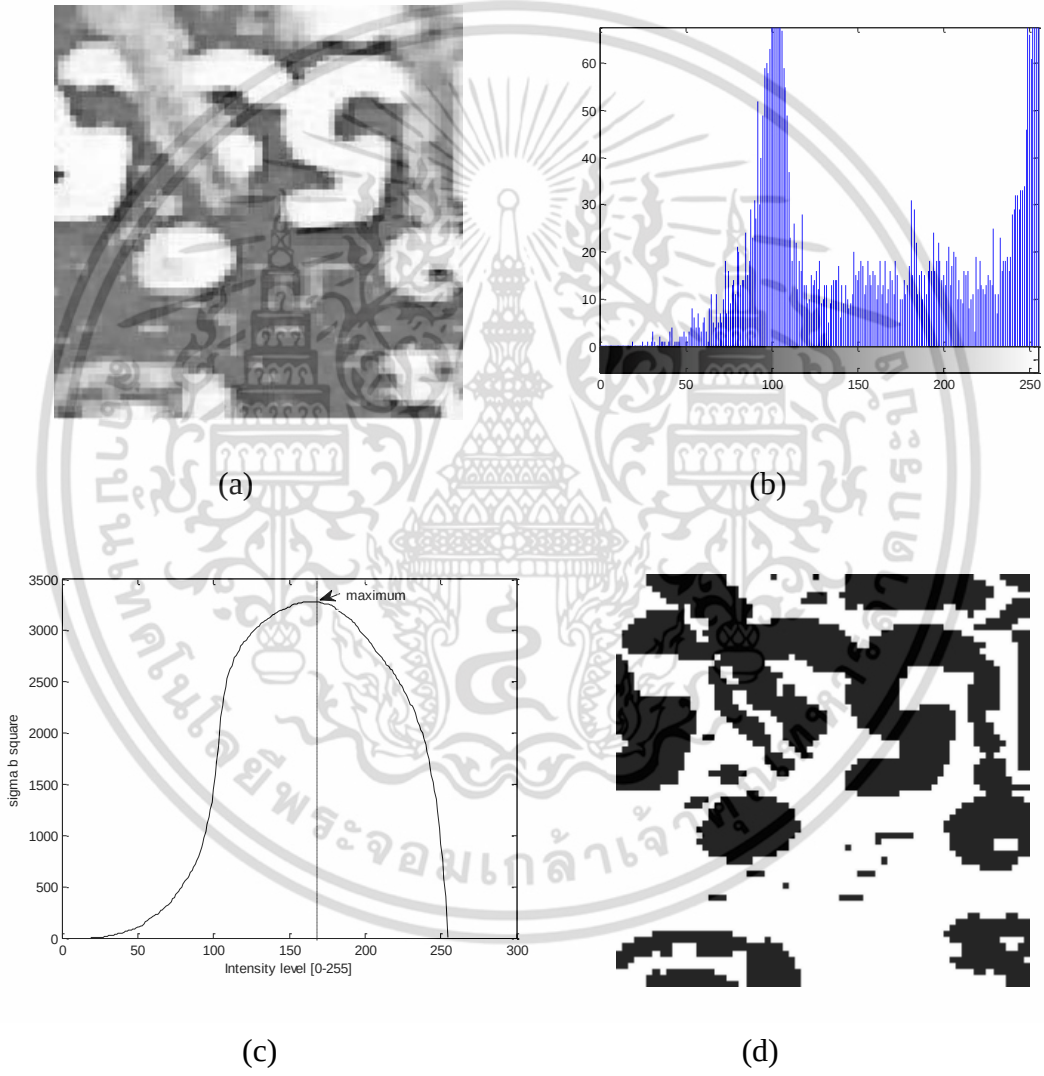


Figure 4.4 Optimum thresholding out of a gray-level histogram using Otsu's method (a) Source Image (b) Histogram of the gray image (c) $\sigma_B^2(k)$ versus gray-level (d) Output image after binarization using the Otsu's thresholding value

$\sigma_B^2(k^*)$=3283 is function of threshold and maximized at the threshold at k^*=165 in gray scale, Fig.4.5(c). T_{otsu}=0.6431 is the optimum threshold that lies in the range [0-1]. The threshold 165 is located at the valley of histogram. Lastly, gray image can be converted to binary using the threshold value, Fig. 4.5(d).

4.3.3 Binary Morphological Techniques

Binary morphology changes the shape and boundary of particles in a given binary image. The binary images may contain unwanted information, such as noise particles, particles touching boundaries of each other. Morphological functions can improve binary images by removing these unwanted particles. The two most common morphological operations are that of dilation and erosion. The binary image and structuring element are represented as the points set, *P* and *S* respectively. Dilation combines the two sets (i.e., *P* and *S*) by using vector addition. The most common structuring element is a 3×3 matrix containing value of 1.

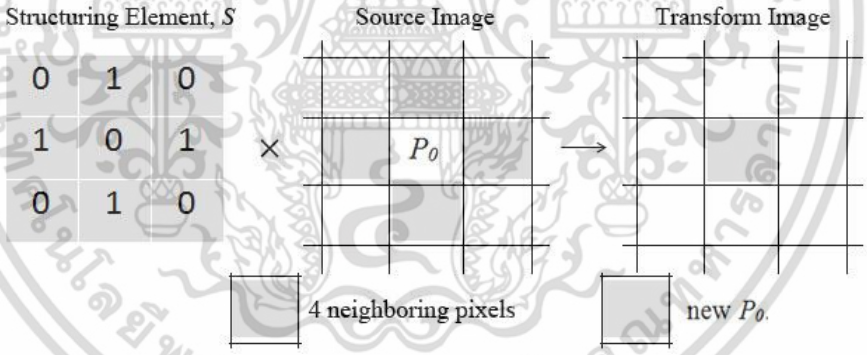
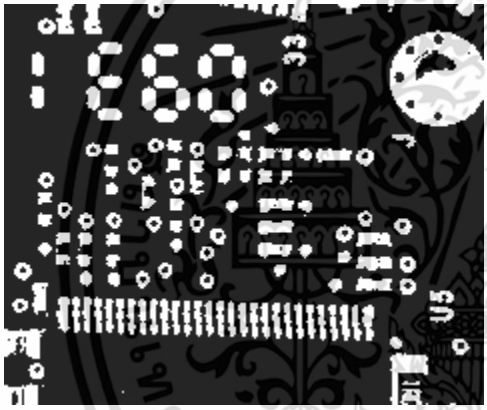


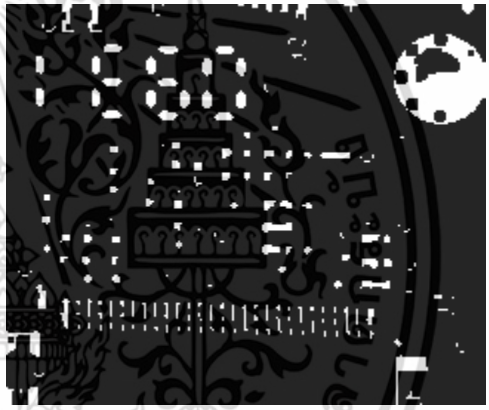
Figure 4.5 Illustrating the use of a 3x3 structuring element on a morphological function

The morphology operations are performed with a structuring mask, *S*, in Fig. 4.5, which is like a filter mask. Fig. 4.5 illustrates a morphological transformation using an *S* to alter a pixel P_0 and its neighboring pixels to become new P_0. The primary morphology functions include three fundamental binary morphological functions; erosion, dilation, and hit-miss. An *erosion* eliminates pixels isolated in the background and erodes the contour of particles according to the template defined by the *S*. A *dilation* eliminates tiny

holes isolated in particles and expands the particle contours according to the template defined by *S*. *Opening* function is an erosion followed by a dilation. It removes small particles and smoothes boundaries. *Closing* function is a dilation followed by erosion. This function fills tiny holes and smoothes boundaries. *Hit-Miss* function locates a particular shape or patterns of pixels. *Thining* function eliminates pixels located in a neighborhood matching a structuring element, *S*. *Thickening* function adds to *P* those pixels located in a neighborhood that matches *S*. Border function removes particles that touch the border of the image. *Hole filling* function fills the holes within particles. Fig. 4.6 shows output of each morphology operation.



(a) Binary source image

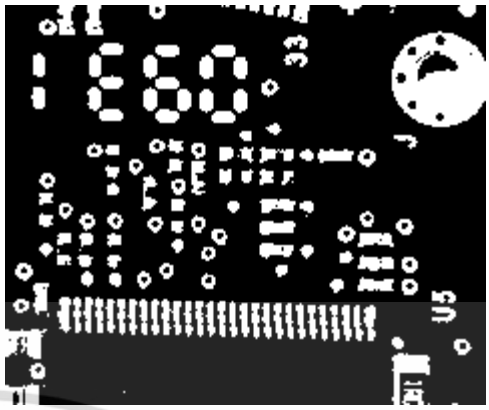


(b) Result after *Erosion*

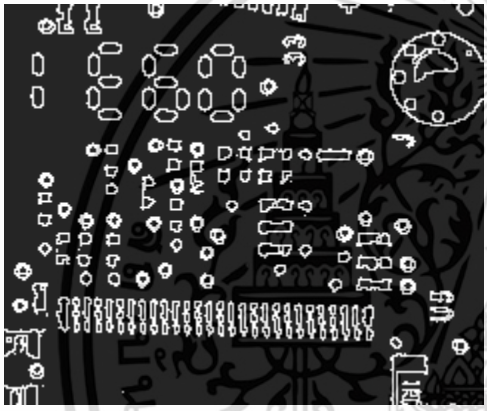
(c) Result after *Dilation*

(d) Result after *Open*

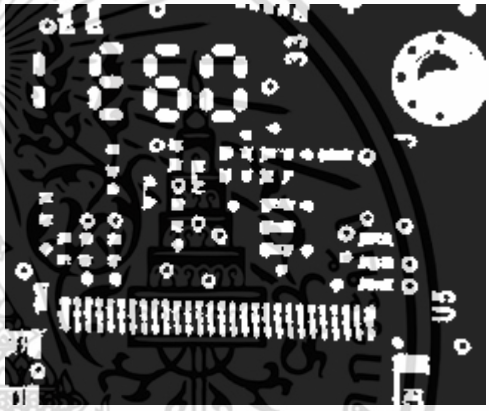
(e) Result after *Close*



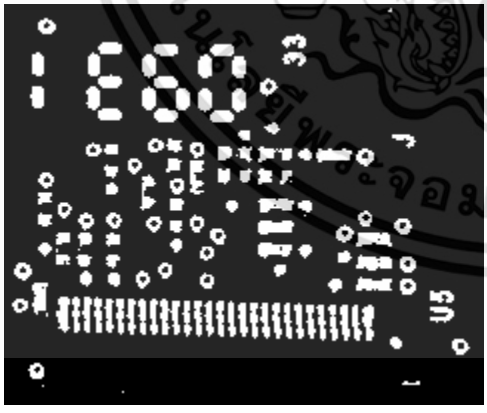
(f) Result after *Thick*



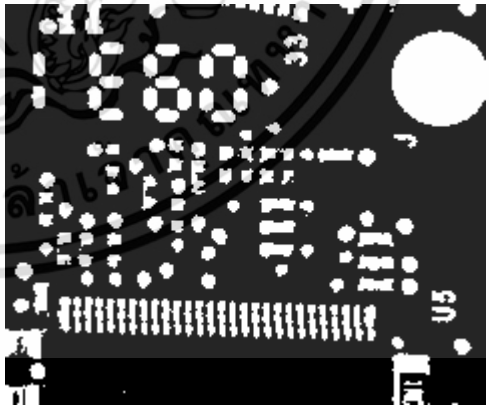
(g) Result after *Thin*



(h) Result after *Remove Small Objects*



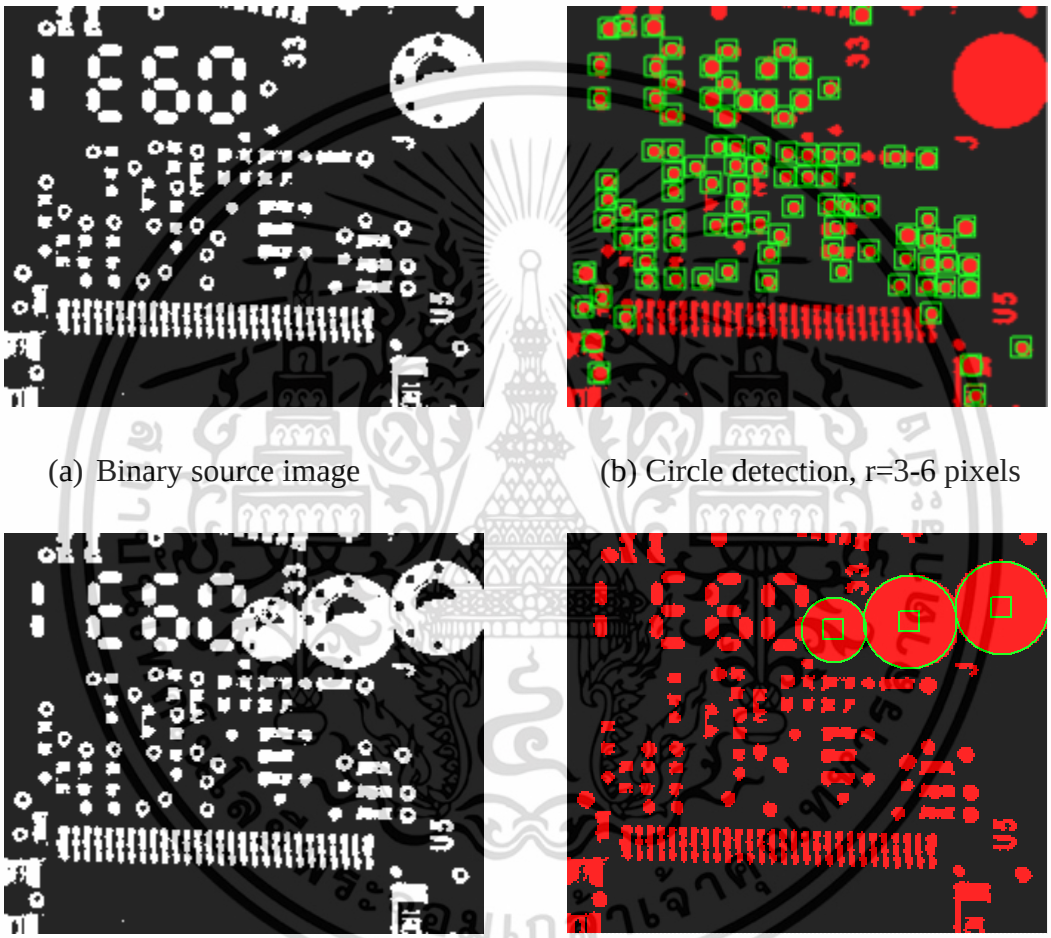
(i) Result after *Remove Border Objects*



(j) Result after *Hole-Filling*

Figure 4.6 Processed images using different Morphological methods

Often, more than one morphological method is applied in sequence according to the source image and desired output before particle analysis. Fig. 4.7 shows a desired output (e.g. circle detection) which requires a combination of multiple morphological steps; *Binarization*, *Hole-filling*, *Dilation/Erosion* and *Hit-Miss* (*circle detection*).



(a) Binary source image (b) Circle detection, r=3-6 pixels

(c) Another binary source image (d) Circle detection, r=15-25 pixels

Figure 4.7 Result of multiple morphological steps for circles detection

4.3.4 Edge Detection

Edge is a point in the image where intensities are changing rapidly between adjacent pixels. Location 4 in Fig. 4.8 represents the rapid change of the intensities. The

point location (*x*,*y*) is called Edge location. The location can be computed with subpixel precision by using an interpolating function. Edge detection finds the edges along a line which is composed of a set of pixels.

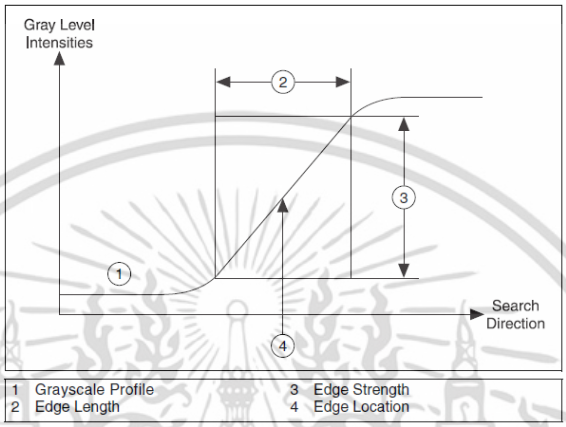


Figure 4.8 Edge model for edge characterization

Characteristics of the intensity changes can be determined for more accurate detection. Edge strength defines the edge contrast which is the minimum difference in grayscale value between the background and the edge. Edge length defines maximum distance in which the desired grayscale difference between the edge and the background must occur.

The intensity changes can be defined by the first derivative of pixel intensity function, *f*, when *f* is a one dimensional intensity profile.

$$\frac{df}{dx} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon}, \tag{4.4}$$

The change of position in image cannot get smaller than 1. Hence, Eq. 4.4 and Eq. 4.5 can be re-written with the first and second derivatives as:

$$\frac{df}{dx} := f(x+1) - f(x), \tag{4.5}$$

$$\frac{d^2 f}{dx^2} := f(x+1) - 2f(x) + (f(x-1), \tag{4.6}$$

The derivative for two dimensional operators can be defined as:

$$G_x = \frac{\partial}{\partial x}\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \text{ and } G_y = \frac{\partial}{\partial x}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \text{ for } x \text{ and } y \text{ direction respectively.}$$

The operator above is most commonly called Sobel operators or Sobel filter masks. Table 4.1 concludes other common masks of edge operators. The Roberts mask are designed to respond to the edges running at 45° while other masks shown in the table respond to the edges running vertically and horizontally.

Table 4.1 Most common masks of edge operators

(a) Roberts 2×2 operator

$$R_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad R_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(b) Sobel 3×3 operator

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

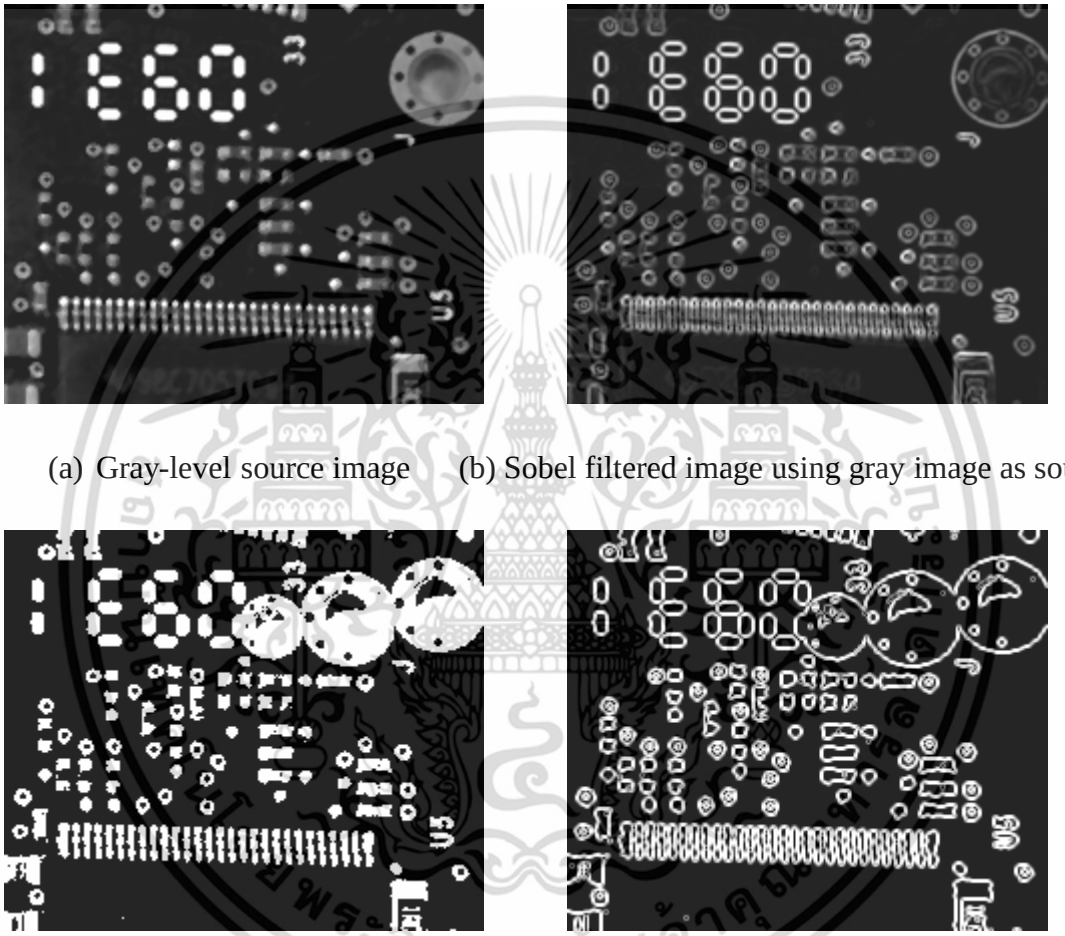
(c) Prewitt 3×3 operator

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

The resulting gradient approximations of rows x and columns y can be combined to give the gradient magnitude, G, as

$$G = |\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = \sqrt{G_x^2 + G_y^2}, \tag{4.7}$$

The gradient magnitude peaks at the locations of the edges in the image. Hence, the edges can be positioned.



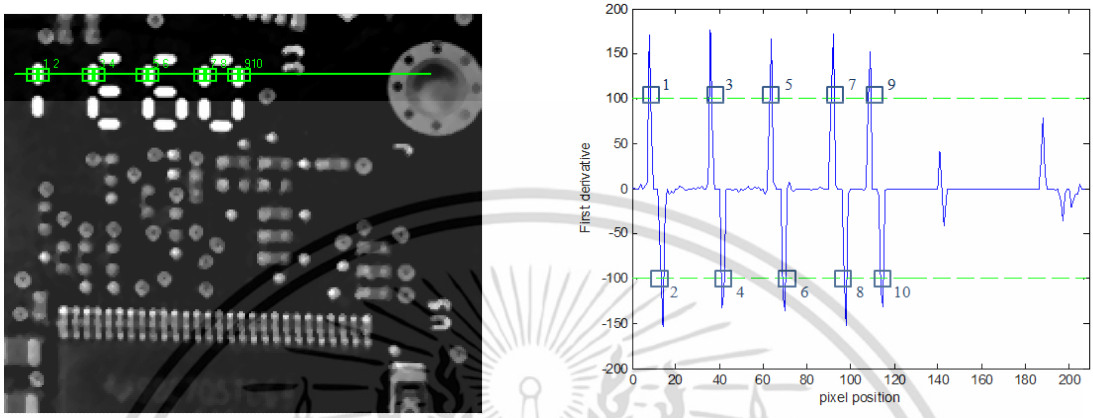
(a) Gray-level source image (b) Sobel filtered image using gray image as source

(c) Binary source image (d) Sobel filtered image using binary image as source

Figure 4.9 Example of filtering source images using Sobel masks

The above derivative operator gives gradient magnitude at each pixel whilst the edge point has not yet been identified. Hence, it needs to apply an analysis to the intensity profile under interest in order to determine the edge point (*x*,*y*). Fig. 4.10 shows an example of a line profile analysis, a line is drawn from the start point, (6,30), to the end

point, (214,30). There are 209 pixels along the line. All edges (i.e., 10) consist of raising edges and falling edges. Minimum edge strength is defined at 100 with bilinear interpolation.



(a) Gray-level source image (b) Intensity profile and edges position

Figure 4.10 Example of intensity profile analysis

4.3.5 Pattern Matching

The pattern matching algorithm is mathematically based on the cross correlation function on a gray image. Cross correlation is a measure of similarity of two sub-pixels. It is commonly used for recognizing objects. In most applications, the known sub-image, *w*, (i.e., template) will be prepared prior in order that it may be compared with the sub-pixels in the acquired image to determine the region and orientation of most similarity.

$$C(i,j)=\sum_{x=0}^{L-1}\sum_{y=0}^{K-1} w(x,y)\,f(x+i,y+j), \tag{4.8}$$

where $w(x,y)$ is a sub-image (golden template) of the size KxL; $f(x,y)$ is the original image of the size MxN where $K \le M$ and $L \le N$. C indicates the matching coefficient value at the pixel position (i,j) where w best matches f.

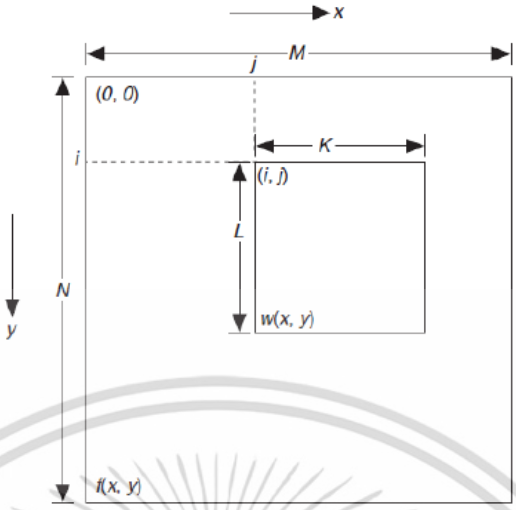


Figure 4.11 Geometric location of an interested object

Fig. 4.11 shows geometric location of an object of interest. Basic cross correlation function is sensitive to the amplitude of gray level changes. Therefore, the cross correlation in normalized form is used to stabilize the amplitude changes. Hence, better precision of correlation can be obtained by subtracting off means so called normalized correlation. The normalized cross-correlation (NCC) can be expressed as:

$$R(i,j) = \frac{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(w(x,y)-\bar{w})(f(x+i,y+j)-\bar{f}(i,j))}{\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(w(x,y)-\bar{w})\right]^{1/2}\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(f(x+i,y+j)-\bar{f}(i,j))\right]^{1/2}} \quad (4.9)$$

In Eq. (4.9) $\bar{w}$ denotes the mean value of intensity of the pixels in the golden template *w*. In the same way, $\bar{f}$ is the mean value of the pixels in the region coincident with the location of *w*. *R* is correlation coefficient which lies in the range -1 to 1. In NI Vision, the best matched is scaled to score between 0 and 1000. Where, 1000 represents a perfect similarity between the sub-image and the template. Fig. 4.12 shows example result of pattern matching. In practice of this work, pattern matching is adopted to locate the region of interest in the image. Then, the pixels in the matched region will require further analysis in order to get better accuracy for the object identification.

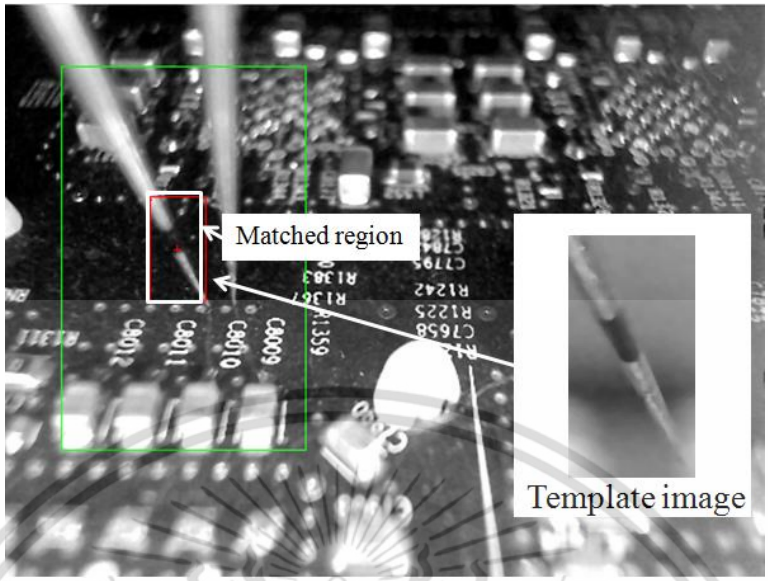


Figure 4.12 Result of Pattern Matching using Normalized Cross-Correlation

A common set of processing steps for pattern matching consist of; image acquisition, convert source image to grayscale image, image smoothing by spatial filters, and then perform pattern matching. The background pixels behind the object under interest can be ignored for a more accurate matching in pattern matching tools. Apart from similarity in the scaled-score, most of pattern matching tools also report; position, orientation and multiple instances of the pattern in the image.

4.4Conclusion

This chapter reviews a number of common image processing techniques, providing the brief background to each technique with examples. Most of the introduced examples are adopted in the proposed visual servo system. A common process of image processing consists of; image acquisition, preprocessing of a grayscale image (i.e., image smoothing or filtering), object segmentation by an intensity thresholding, morphological techniques and object interpretation. Often, feature, line, edge and point analysis are applied to grayscale image. Although some of the important techniques are not included in this chapter.

CHAPTER 5
FEATURE EXTRACTION: PROBE TIP ENDPOINT AND TEST POINT IDENTIFICATION

5.1Introduction

This chapter provides details on the image feature extractions. The features of probe tip and test point are focused as the target objects for further identification. The solutions of the feature extraction are implemented in the proposed visual feedback. The discussed solutions aim to solve vision issues of the implemented robot application. The visual servo can be categorized as the Position-Based Visual Servo (PBVS) as described in chapter 2. Fig.5.1 shows the general workflow of the PBVS. The visual servo is integrated into an automated test system as depicted in Fig. 5.2 [18-20].

Most automated testers used in manufacturing require reliability and exactness. Many of today's commercialized visual machines work only under a controlled environment with a limited number of target objects for its vision system. In contrast, the target objects for visual detection (i.e., test point on PCB) cannot be completely categorized because of the fact that there is no clearly defined specification existing for carrying out feature extraction. It is not cost-effective to designate part of the PCB footprint as the test point separately from the main design to accommodate visual sensing. As a result, various kinds of test points have been implanted into PCB industrial design.

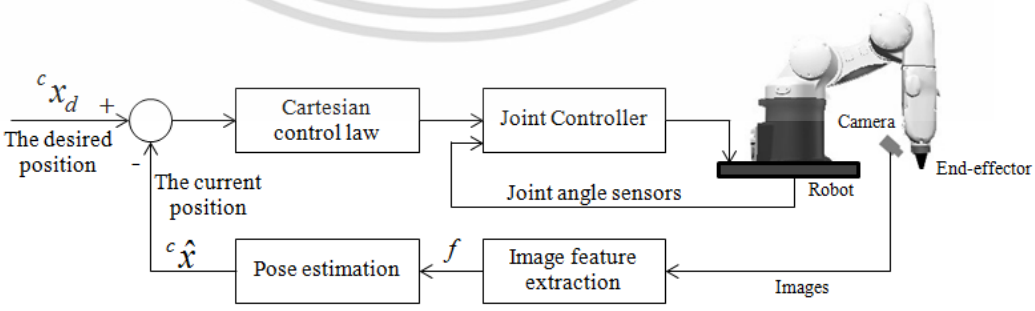


Figure 5.1 The structure of a position-based visual servo (PBVS).

5.2Visual Closed-Loop for Solving Robotic Issues in Manufacturing

The proposed PBVS is applied to be a feedback loop of a Cartesian robot that servos an end-effector to the test point on PCB. The problem with the robot is sometimes can incorrectly place the end-effector in relation to the target (i.e., test point) on a printed circuit board (PCB). Therefore, a correction of the end-effector placement is required. Visual servo has been designed for the pose correction of the end-effector.

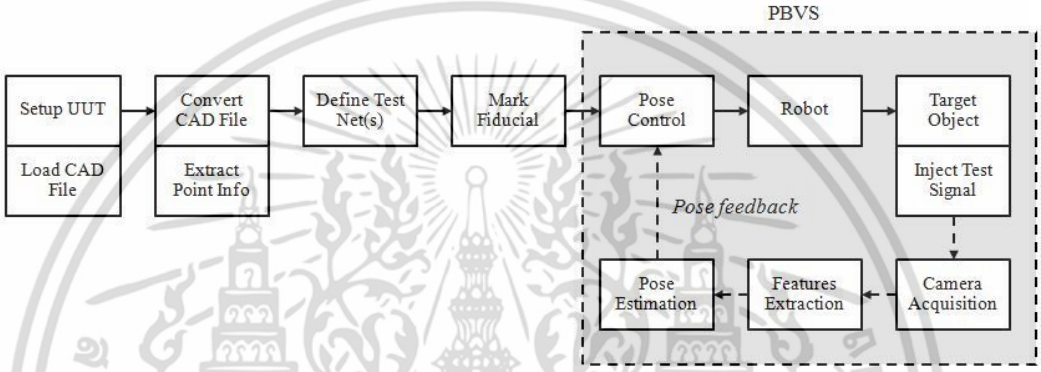


Figure 5.2 System workflow of an implemented PBVS for a Cartesian robot.

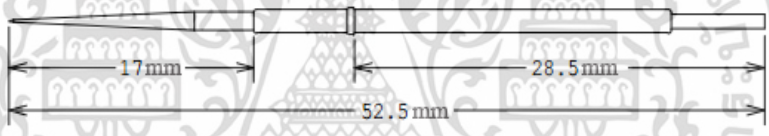
The system workflow is depicted in Fig. 5.2, a visual servo is integrated into an automated tester. First, an operator loads the Unit Under Test (UUT) into the tester. The software program extracts test points and plots in the memory the coordinates of the test points. The operator selects the nets (i.e., test points) to test and defines the test method for each test point. The operator guides the robot to three fiducial points. The robot servos the probe to the first point and with relative distance refers to the selected fiducial points. Then, the robot inserts the test signal. The computer software collects the test results for further analysis. Nevertheless, the robot opens its eye (i.e., camera) to investigate the pose if it senses an abnormal signal, which is potentially caused by an open circuit because the probe tip does not properly touch the conductive pad.

Since the visual servo mimics the human sense of vision, the feedback loop is activated when a machine error occurs. The proposed feedback loop tracks both the target object and the robot's end-effector. Target objects are test points which are mostly

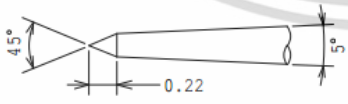
connective pads on PCB. The end-effector is a pin probe that has a needle end. The probe is attached with the rigid body of the Cartesian robot. In the next section the algorithm for identification of the probe will be introduced.

5.3Probe Tip Endpoint Identification

The body of probe can be easily identified by using pattern matching. Fig. 5.3 shows the geometry of the probe, body and its end. Identification of its endpoint solicits more effort as it requires a custom algorithm. At a robot position, the loop acquires two images using the white light to allow extraction of the probe tip endpoint, x_e, and then the red light for the test point, x_g, identification.



(a) Size of Pogo Probe Tip



(b) Tip-end, needle type (c) Compare size of the probe with a coin

Figure 5.3 Pogo pin specifications

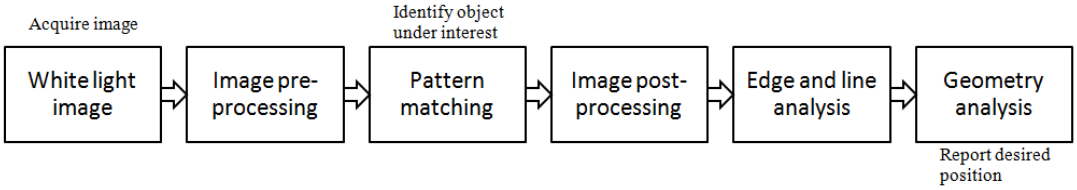
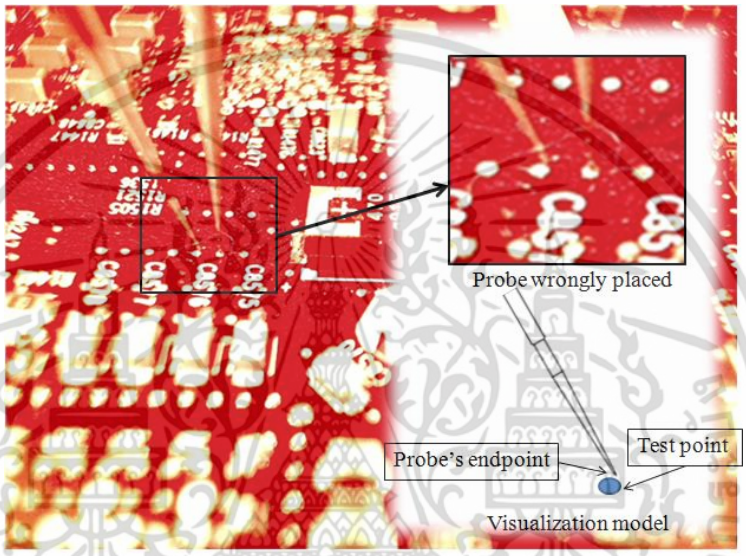
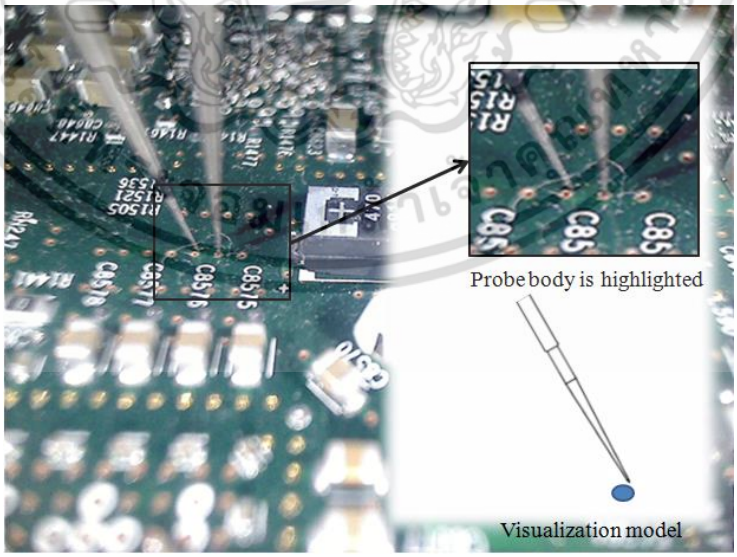


Figure 5.4 Image processing steps toward identifying the endpoint of the probe



(a) Image acquired with red light



(b) Image acquired with white light

Figure 5.5 Images acquired with different light color (a) red light (b) white light

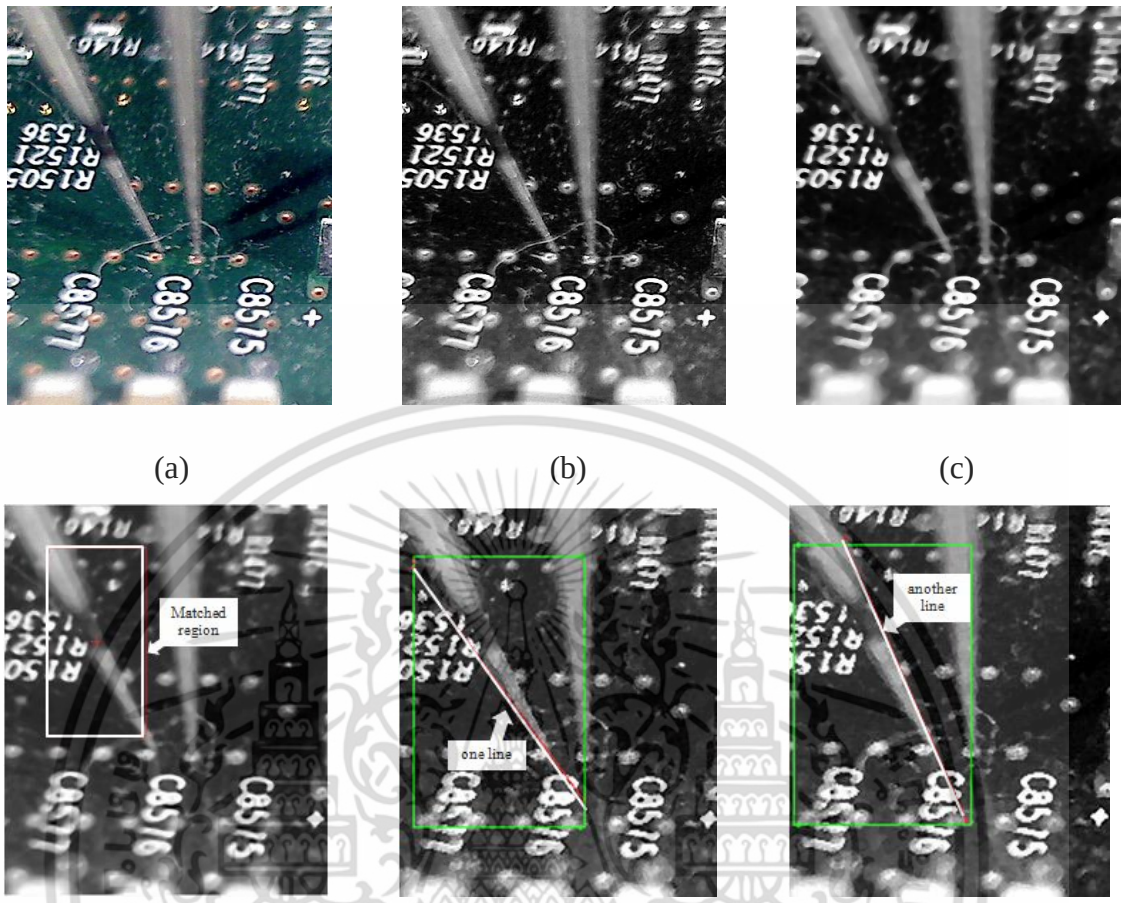
Fig. 5.4 shows processing steps of tip's endpoint detection starting from acquiring image towards geometry analysis. The steps are: (1) Once the visual servo is triggered, the software instructs the controller to turn on white light and then acquire an image, (2) the acquired image is pre-processed using preprocessing techniques described in chapter 4, the output is gray image at this step, (3) pattern matching using normalized cross-correlation function, (4) edge of object of interest is highlight at pose-processing, (5) extract edge and line out of the gray image, (6) geometry of the detected edges and lines are analyzed. The final output is the position of tip's endpoint in image coordinates. White light and red light are used while acquiring the image which will be used for the endpoint extraction and test point extraction respectively [21].

5.3.1 Image acquisition

Target object from the camera view can be highlighted by proper reflection of the light source. An appropriate light source with correct position enhances edge visibility and contrast between the object under interest and the background [21]. Pogo pin probe is made of metal which normally does not absorb light. The probe reflects light in all directions. Hence, white light is turned on at this step of image acquisition using a Logitech camera (model C600). In the implemented process, the sources of red and white light are switched by a controller for acquisitions of test point and pin probe respectively. Fig. 5.5(a)-(b) show the images which are acquired with red light and white light respectively.

5.3.2 Image pre-processing

The original color image, Fig. 5.6(a), is converted to a gray scale image, Fig. 5.6(b). Often, images are corrupted by random variations in intensity values. Some common types of noise are salt and pepper noise, impulse noise, and Gaussian noise. The converted gray image is smoothed by pre-processing (i.e., median filtering) to have the gray image with low noise for further pixel analysis. A filter size, 7×7, is applied as the median filter parameter, Fig. 5.6(c). Median filters are very effective in removing the typical noises (e.g., salt and pepper noise) while preserving image details (i.e., edges are minimally degraded).



(a) (b) (c)

(d) (e) (f)

Figure 5.6 Result of image processing steps (a) original image cropped near interest region (b) gray image (c) pre-processed image (d) pattern matching (e)-(f) edges detection

A median filter which is the simplest nonlinear filter which can be denoted as:

$$g(i,j) = \underset{-k\le u,v\le k}{\text{med}} \left\{ f(x+u, y+v) \right\}, \tag{5.1}$$

where the median is calculated over a $(2k+1) \times (2k+1)$ window. The window is equal in size to the filter.

5.3.3 Pattern matching

Pattern matching algorithms are mathematically based on the cross correlation function on the gray image, as described in chapter 4.

$$C(i,j)=\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}w(x,y)\,f(x+i,y+j), \tag{5.2}$$

$$R(i,j)=\frac{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(w(x,y)-\overline{w})(f(x+i,y+j)-\overline{f}(i,j))}{\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(w(x,y)-\overline{w})\right]^{1/2}\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(f(x+i,y+j)-\overline{f}(i,j))\right]^{1/2}} \tag{5.3}$$

(a) (b) (c) (d)

Figure 5.7 Results of Pattern Matching with different scores, (a) Template image 220×116 pixels (b) 847.97 match (c) 855.40 match and (d) 851.69 match

where $w(x,y)$ is a golden template of the size $K\times L$; $f(x,y)$ is the original image of the size $M\times N$ where $K \le M$ and $L \le N$. C indicates the matching coefficient value at pixel position (i,j) where w best matches the gray image f, in Eq. 5.2. While $R(i,j)$, in Eq. 5.3, represents the correlation coefficient in moralized form. The normalized correlation

results give better precision of the best match against variation of the amplitude changes of intensity level in the original gray image f .

5.3.4 Image post-processing

Frequently, a suitable gray image for the cross correlation may not be suitable for the edge and line detection algorithms. Thus, post processing is applied to the gray image to enhance the interested detail in the image in order for it to stand out for easier detection. In this example case, edge and line are interested for further geometry analysis. Hence, a convolution function with kernel 3×3 is applied to highlight the bright and dark objects in the image to facilitate the edge and line detection.

$$\begin{bmatrix} h4 & h3 & h2 \\ h5 & h0 & h1 \\ h6 & h7 & h8 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure 5.8 Convolution filter mask for intensity highlight

Fig. 5.8 shows a convolution filter mask for intensity highlight. The filtered value of a pixel is a linear weighted combination of its original value and the values of its neighboring pixels. The neighboring pixels' scheme are commonly numbered as: [P4, P3,P2;P5,P0,P1;P6,P7,P8].

$$\text{CONVOLV}:[[Q0 := P0\times h0 + P1\times h1 + P2\times h2 + P3\times h3 + P4\times h4 + P5\times h5 + P6\times h6 + P7\times h7 + P8\times h8]]; \quad (5.4)$$

Thus, edges and lines are highlighted after operation of the convolution filter, as shown in the output image Fig. 5.6(e)-(f). The output image is sharpened for further step of line

detection. Fig. 5.9 shows an example of computation of the new filtered value (e.g., 112) at *P*0[101,101]: (108×10)+ (85×(-1))+ (82×(-1))+ (96×(-1))+ (118×(-1))+ (128×(-1))+ (131×(-1))+ (120×(-1))+ (96×(-1))=224 the result of which is divided by the sum of the weights (i.e., 2).

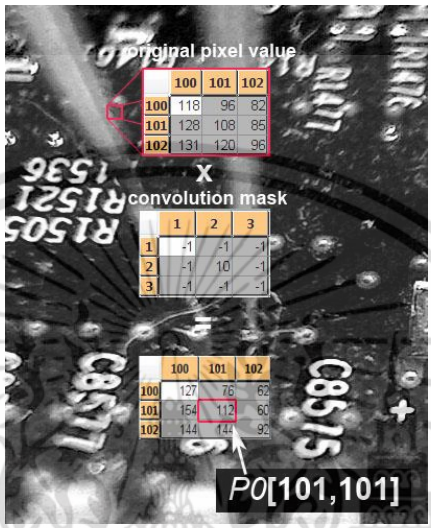


Figure 5.9 Example computation of a filtered value using a convolution filter

5.3.5 Probe's Edge and Line Analysis

Interestingly, some poor fits during pattern matching returned the relatively high score of 848 producing a wrong orientation. As a result, the detection of the edges and lines are implemented for more accurate identification of the endpoint. Two straight lines in the matched region, as shown in Fig. 5.6(e) and 5.6(f) respectively, are obtained using an edge detection tool in NI (i.e., IMAQ Find Straight Edge). Lines L_1 and L_2, in the directions from right to left and left to right, are detected respectively. Frequently, x_i is mistakenly regarded as the end point despite the fact that the angle of the two lines is subject to lighting conditions.

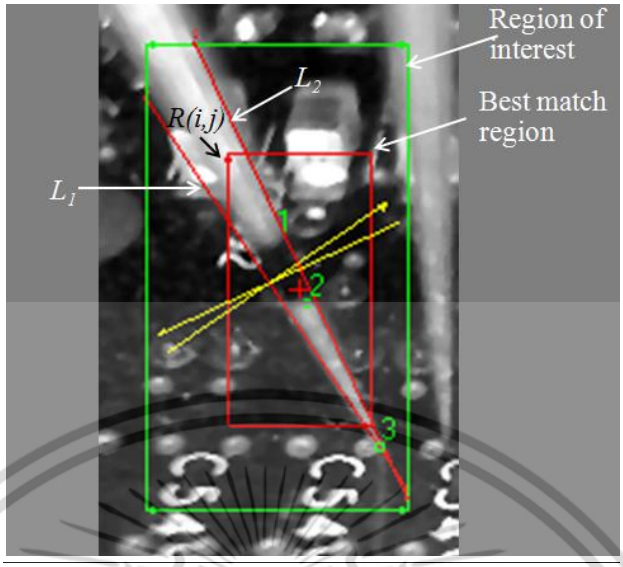


Figure 5.10 Example result of lines and edges detection inside the best match region

Therefore, geometry calculation of lines and points in the next processing step are applied. In Vision Assistant, the main image processing steps can be illustrated to facilitate further coding in NI LabView or other such programming languages. Fig. 5.10 shows the two detected lines (e.g. L_1 and L_2). Geometry analysis of the edges and lines is the next step for positioning the tip's endpoint.

5.3.6 Probe's Geometry Analysis

In Fig. 5.10, lines L_1 and L_2, in the directions from right to left and left to right respectively for directional extraction, are located first prior to determining the intersection point x_i.

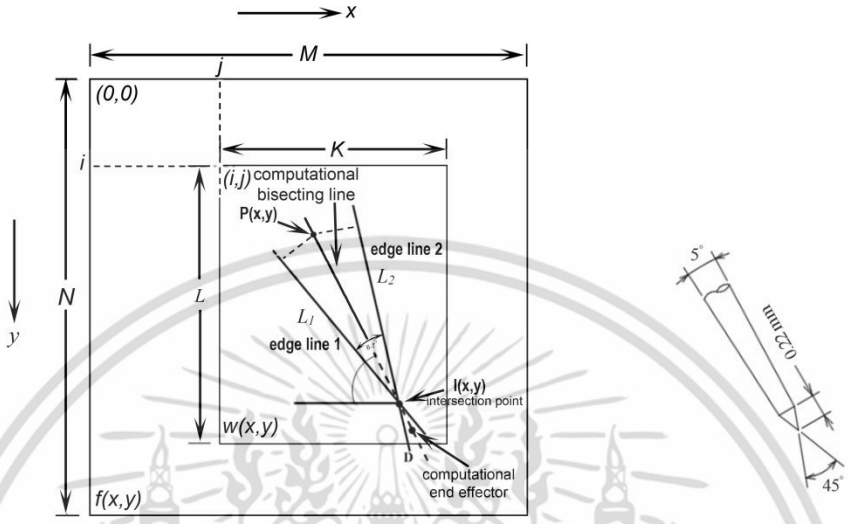
$$f_{L_1}(x,y) \equiv a_1 x + b_1 y - c_1 = 0 \tag{5.5}$$

and

$$f_{L_2}(x,y) \equiv a_2 x + b_2 y - c_2 = 0 \tag{5.6}$$

Therefore, the intersection point, $x_{i,}$ is:

$$x_i = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{5.7}$$



(a) (b)

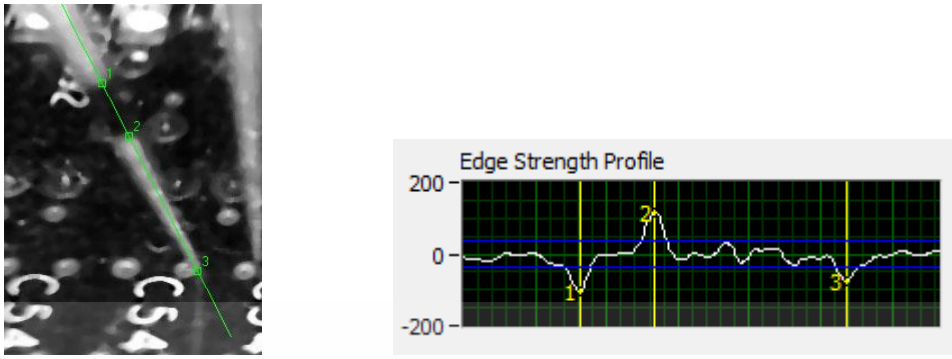
Figure 5.11 (a) Geometry coordinate of cross correlation and (b) Needle pin-end characteristics

Angles of the two straight lines can be changed due to lighting conditions. Therefore, the tip endpoint will be obtained by calculation instead of relying on the lines detection and their intersection.

$$X_{e,i} = \{(i, j) \in Z : p_x \le i \le i_x \wedge p_y \le j \le | x_{e,i} - x_{e,0} | \tan\theta \tag{5.8}$$

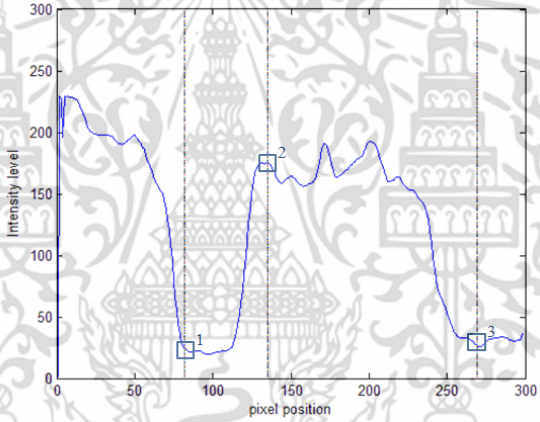
$$x_e = (x_{end}, y_{end}) = (x_i + L\cos\theta, y_i + L\sin\theta) \subset X_{e,i} \in Z \tag{5.9}$$

where $X_{e,i}$ is a set of pixel elements along the bisecting line. x_x is the endpoint after the bisecting line extension to ensure the bisecting line is passing through the actual pin's endpoint. On the other hand, x_x can also be described as an extension to the intersection point x_i.

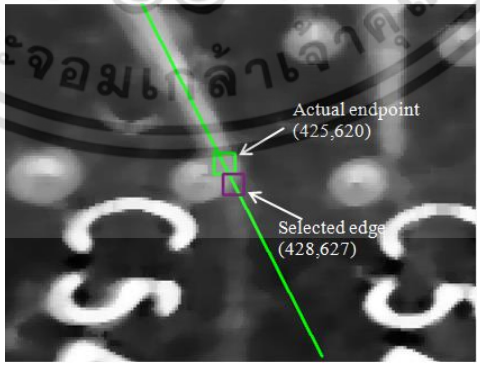


(a) (b)

Figure 5.12 Analysis of pixels intensity (a) line from start to an extended point (b) its corresponding edge strength profile



(a) Intensity of pixel elements along the bisecting line and detected edges



(b) Detected edge and new calculated endpoint

Figure 5.13 Pixel intensity and position of the detected edges

Along the bisecting line, x_e is the endpoint which is an element in the subset of $X_{e,i}$. x_p is the intersection point of a perpendicular line to L_1 and another to L_2. A straight line is then drawn between the point x_p and x_x as shown in Fig. 5.12. The line profile is analyzed using the IMAQ Edge Detector of bilinear fixed interpolation type for intensity variations along the bisecting line $X_{e,i}$. Points 1 and 2 are the sharp edges representing the upper and lower rims of the black marking. Due to constraints in the development time and the lengthy amount of time required if an order for a custom pogo pin were placed from the manufacturer, the black marking was manually made on to the pin tip to increase visibility of the line profile as the sharp edges of the manual marking can satisfactorily produce accuracy in real applications. The clear instructions of the manual marking procedures should be made available to the workers as the imperfections of the manual marking will be removed by the automatic calibration process. The practices of manual marking and calibration are a frequently utilized solution in solving urgent manufacturing problems under time pressure.

The distance from the second rim along the bisecting line to the end-effector endpoint is h pixels. h is determined when the operator's selects an endpoint on the computer screen during the fine-tuning process. The endpoint should be positioned against the low noise background, e.g., black color background. In Fig. 5.12a, The probe is clearly visible however an accurate identification of the probe endpoint by visual inspection requires great effort. The point coordinates, x_p=(306,388) and x_x=(458,685), are the start and end points of the bisecting line. The line profile is the intensities of all pixels on the bisecting line, shown in Fig. 5.13a. The second rim of the marking is at (367,507) and its distance (134^{th} pixel) to the endpoint is 127 pixels, giving the endpoint of end-effector at (425,620). Fig. 5.13b shows that the new calculated endpoint is more accurate than considering the 3^{rd} edge as the endpoint. The algorithm is thereafter called the Tip Endpoint Detection Algorithm (TEDA).

5.4 Test Point Identification

Test point is another interest target object in this thesis. The visual servo must be able to provide feedback to the robot in order for correct specific placement of the end-effector onto a test point. Therefore, test point detection is one major challenge in the design of software. In the software design, detection of various types and shapes must be taken into account. Thus, this work attempts to design a software program that can facilitate both current and future algorithms to allow for any supplementary algorithms that can be added as needed to meet the new requirements.

The test-point detection algorithms are classified into two categories: One is to confirm and the other to exclude the test points from the possible list. The scoring technique for test point identification is not utilized in this work so as not to mislead the operator with the high and low scores of the objects under examination. Instead, the proposed software program would cease the signal insertion at the test point if too many identification constraints exist, the phenomenon which requires additional advanced algorithms for greater extraction capability. The unknown test points, due to the constraints, are arranged in sets of pixel elements bounded by the regions of interest (ROI array) and then analyzed. The software architecture is termed the Take and Throw (TaT) algorithm.

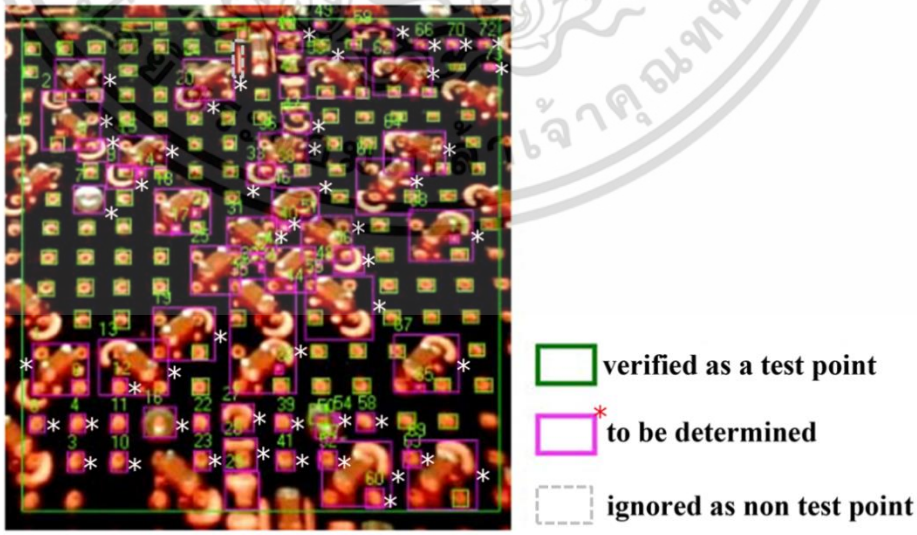


Figure 5.14 Feature classifications by the TaT algorithm

Following the preprocessing steps, the images are segmented into two regions: the object region and the background region. In segmentation, the multiple-class clustering technique of the NI Vision software was applied to various light intensities [22]. A set of $k=\{P_1,P_2,\ldots,Pn\}\in Z^2$ represents the number of ROI's of the segmented objects, each of which is a subspace of the large outer ROI, and $P_i=P_i(p_t,p_b)$ denotes an ROI block of rectangular type, each of which consists of the top-left corner pixel at $p_t=(x_t,y_t)$ and the right-bottom pixel at $p_b=(x_b,y_b)$.

The classification process of the objects utilizes the sub-algorithms of the TaT architecture. As seen in Fig. 5.14, the *take* (i.e., the first T) accepts the objects when the criteria are met, whereas the *throw* (i.e., the second T) discards the objects when they fail to meet the criteria. An object that belongs to neither the *take* nor the *throw* is assigned the group under considering. With these programming features, the next generations of software developers are able to add multiple algorithms to the existing sequences to increase identification coverage.

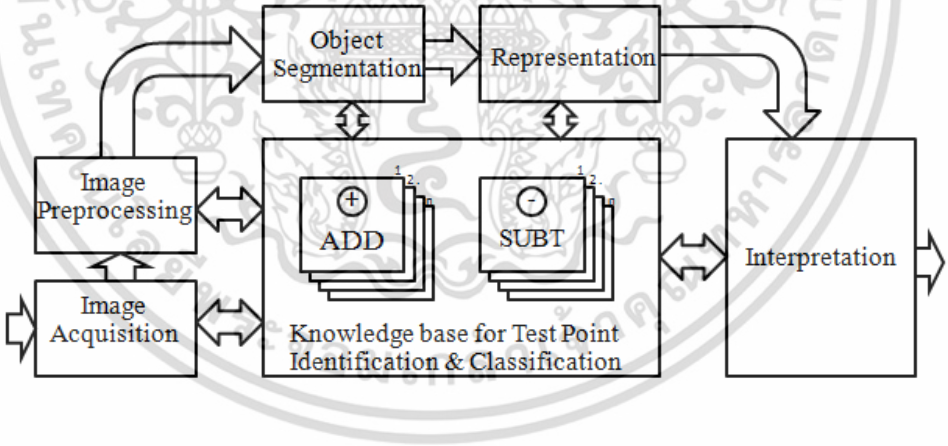
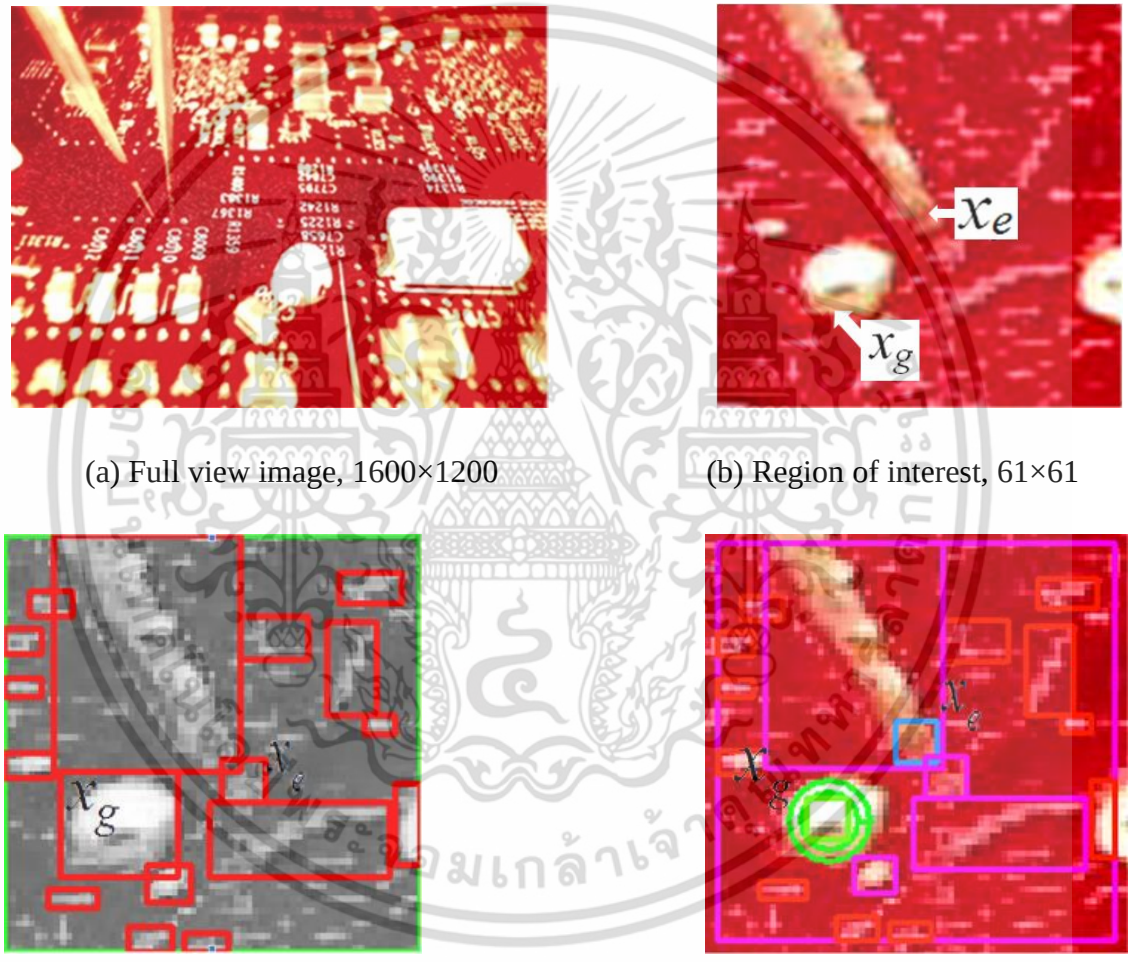


Figure 5.15 The take and throw algorithms for image feature extraction

Fig. 5.15 illustrates the positions of multiple algorithms in the image processing, where ADD and SUBT denote *take* and *throw*, respectively. The remaining ROI's that belong to neither group are to be assigned to the *to be determined* (TBD) area pending the development of more advanced algorithms. PBVS will return significant FAULT to the

system for PBVS that falls outside the recognition scope. The TaT method is similar to the Not_Target and Target_State method by which the transition was tallied from total votes [6]. However, since exact identification is required and the voting method of Not_Target and Target_State fails to offer such exactness, the TaT approach is thus selected.



(a) Full view image, 1600×1200 (b) Region of interest, 61×61

(c) Object segmented, 61×61 (d) Identify objects, 61×61

Figure 5.16 Outputs in the region of interest

In practice, the first-tier ROI, O_c, is enlarged by growing the area in all directions near the probe tip endpoint x_e. The endpoint coordinates are obtained using the TEDA algorithm as described in sub-section 5.3.

$$O_c = \{(x_t, x_b) \in Z^2 : \ x_t = x_e(i-m, j), \ x_b = x_e(i, j+n)\}, \tag{5.10}$$

$$p_c = [p_{c,1} \cdots p_{c,k}] : p_i = \sum_i \sum_j I(i, j) \subset O_i : O_i \subset O_c \in Z^2, \tag{5.11}$$

$$x_g = center(O_t) : O_t = \underset{p_{c,i}}{\overset{k}{\wedge}} (| x_e - center(p_{c,i})_{(x,y)} |), \tag{5.12}$$

$$O_i = O_i^{p_i} : \min \| e \| : e = p_c - p_{ref,i}, \tag{5.13}$$

Given that $O_c = Z_m \times Z_n$ and O_c is enlarged by 61×61 pixels (Fig.5.16), $m=n=30$. O_c consists of a set of ROI elements, O_i , where i represents the number of objects bounded in O_c, all of which can be expressed as $O_c = (O_1, O_2, O_3, \ldots, O_n) : O_i \subset O_c$. Each object ($O_i$) in binary subspace is passed through the TaT algorithm to determine the likelihood of being a test point. The selection of 61×61 pixels for the ROI enlargement is in response to the possible robot error during the machine calibration and to the size of the largest test point on the PCB. Fig. 5.16a is the 1600×1200 resolution full view camera image while Fig. 5.16b is the image of the area inside O_c.

In selection of a test point, the point with its center at x_g and closest to the probe tip endpoint is chosen. The 61×61 pixel windowing is used for the image processing to minimize the number of processing pixels. The above algorithm is termed the Test point Detection Algorithm (TDEA). After the robot has completely stopped at the position ${}^wP(x,y,z)$, the software instructs the controller to switch on the white light and run TEDA to obtain x_e. While motionless, the robot instructs the controller to switch on the red light and acquire an image as an input into TDEA to get x_g.

5.4.1 Test point classification

The review of existing literature has revealed that no attempt has been made to classify the test points. As a result, this thesis classifies test points into two main groups: the via hole and the probe site. The via hole consists of a pad and a hole with respective diameters of approximately 150 and 50 microns, while the probe site has the diameter of 600 microns [23].

As seen in Fig. 5.17, in Figs. 5.17a and 5.17b the test points are readily detectable and of round shape with a dark aperture. Nevertheless, the via in Fig. 5.17c contains no small dark aperture. Figs. 5.17d, 5.17e and 5.17f show the test points belonging to the probe site group. Detection of the probe site requires less effort than the via detection because of the larger size of the probe site. Fig 5.17g consists of two test points, tangent to each other at the white-colored outer edges. The algorithm applicable to Fig. 5.17g must be able to divide the two test points into two separate test points.

(a) (b) (c) (d) (e) (f) (g) (h)

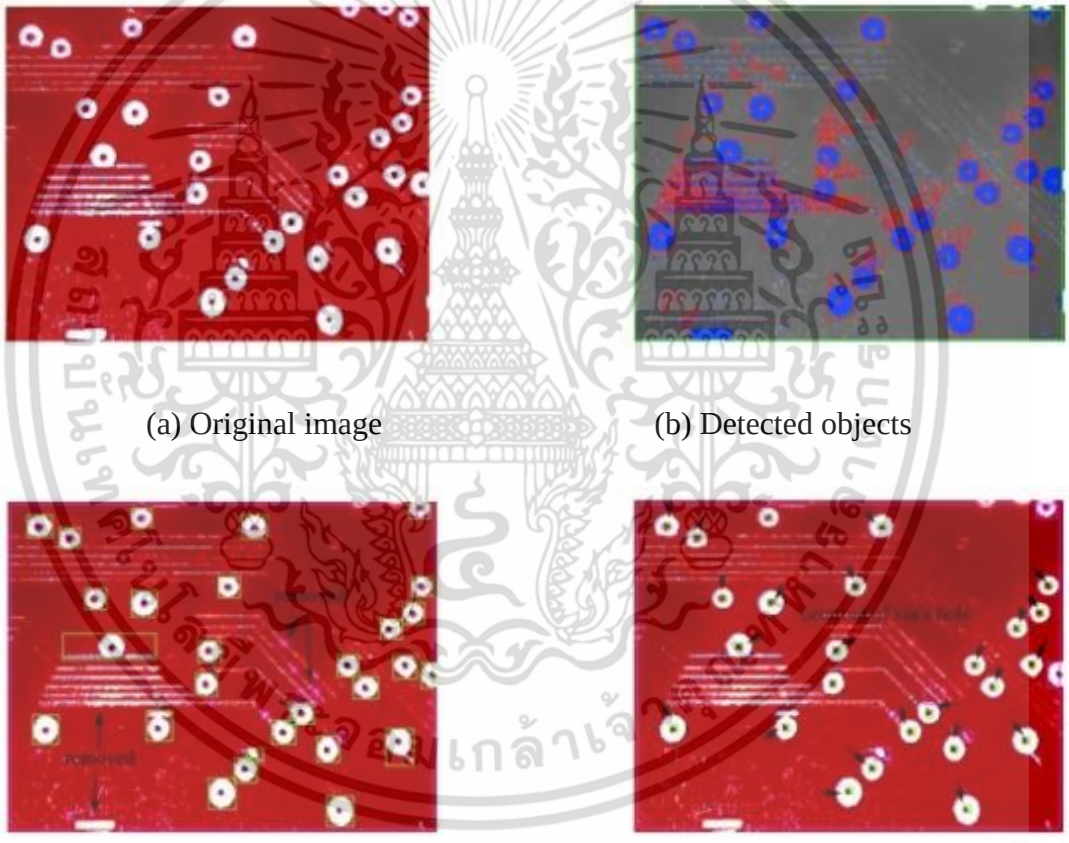
Figure 5.17 Examples of test points in gray

The test point in Fig. 5.17h is a via connected to a net. These test points (Figs 5.17a – 5.17h) are typically found in modern PCBs. Some test points contain a hole in the center even though the hole does not appear in the acquired images as it has been filled by solder from the Surface Mount Technology (SMT) machine. Ideally, the holes of both groups of test points, i.e., the via and the probe site, should be located at the center of the circular pad. In reality, the hole is however shifted slightly to the edge due to a drilling

error [23] as depicted in Fig. 5.16b. In addition, the camera projective effect could make the hole appear slightly off the center.

5.4.2 Algorithm Design for Proper Via Detection

A circular conductive pad with a hole in its center is ideal for the detection algorithm design. The pad is in white as light is reflected whereas its hole is in black, as seen, e.g. in Figs. 5.18a, 5.18b.



(a) Original image (b) Detected objects

(c) ROI's after screening (d) Detected objects

Figure 5.18 Ideal via with black hole extraction

The detection process is as follows:

1) The original image is taken with red light as shown in Fig. 5.18a, in which the interested area is cropped from the original image;
2) The original image is converted to the gray image using the Hue Saturation Intensity (HSI) model, median filter and automatic threshold clustering techniques;
3) Segment the bright object from the background using NI Vision Builder AI (Fig. 5.18b);
4) Remove thin ROI's, which are defined as those with the width less than 2 and the height more than 10 pixels; in other words, with the ratio of width to height larger than 1:5 (Fig. 5.18c);
5) Remove small objects (<20 pixels) and accept ROI's with an area equal or larger than 20 pixels for the next step (Fig. 5.18c);
6) Select the circular objects from the subspace in the remaining ROI's using the Heywood algorithm.
7) Attempt, if possible, to fit a circle with the radius between 5 to 8 pixels into the objects using the IMAQ Find Circles, accept those objects that the circle fits and determine the center of the fitting circle;
8) Locate the dark spot on the bright object using the IMAQ Remove Border Objects (Fig. 5.18d);
9) Analyze the remaining objects and assign a hole or the dark spot as the center of the test point (Fig. 5.18d). However, if the dark spot is unable to be located, deploy the center of the fitting circle of 7);
10) Transform the sub-pixel position of the center location to the image plane coordinates.

The software program expects the vias to have a radius of $5 \le r \le 8$ pixels. Therefore, the test point area is between $20 \le r \le 51$ pixels. Following the 10-step process, the algorithm can eliminate many unwanted objects. As shown in Fig. 5.16d, the unwanted objects are in the small red rectangles and most of the unwanted objects are lines and noise. The

unwanted objects are discarded and labeled as non-test points by the SUBT algorithm. The SUBT algorithm first checks the thinness of each object and if ROI of the object is less than 5 pixels. Any objects that do not come up to specification will not be accepted as a test point even though the area meets the criteria.

5.4.3 Algorithm Design for Probe Site Detection

A probe site area is larger than a via with the former's radius of $14 \le r \le 21$ pixels. Fig. 5.19 shows the detection outputs starting from the original object (Fig. 5.19a) to the center extraction (Fig. 5.19f). The object is processed by the ADD algorithm. By visual inspection, the ADD algorithm should accept the object in Fig. 5.19a as a test point and then correctly locate the center of the object. The algorithm then assigns the dark point at the center as the hole or object center. In the case that the dark hole is not visible on the image, the algorithm computes the average coordinates to be alternatively used as the center of object.

(a) (b) (c) (d) (e) (f)

Figure 5.19 Detection outputs of test points with a black hole

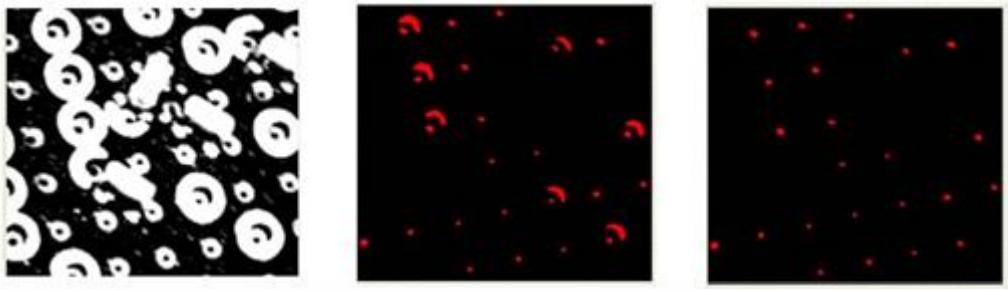
The procedure of image processing underneath is included in the software:

1) The original image is acquired with red light (Fig. 5.19a);
2) The original image is converted to the gray image using the Hue Saturation Intensity (HSI) model and median filter techniques;
3) Detect the objects using NI Vision Buider AI (Fig 5.19b);
4) Remove the small and thin objects with thin ROI;
5) Segment the objects using the automatic threshold clustering method and look for the bright objects (Fig. 5.19c);
6) Detect and select the large circular objects;

7) Invert the binary objects (black and white are reversed, Fig. 5.19d);
8) Remove the border objects (Fig. 5.19e);
9) Analyze the single remaining object and identify the center of test point (Fig. 5.19f);
10) Convert the subpixel position into the image plane coordinates.

The algorithms in the TaT architecture in sub-sections 5.4.1 and 5.4.2 are the knowledge base of test point recognition. As seen in Fig. 5.20, the extraction results using the TaT algorithms show that the algorithms recognize the test point types and that the knowledge base is able to confirm the functionality of the visual identification.

In addition, Fig. 5.20 indicates the need for a greater variety of input images in the software verification. Therefore, the first-tier ROI is enlarged to entail a greater number of image objects for verification, as shown in Fig. 5.20c. Although the expanded window size is recommended for software verification, the reduction of window size is adopted in the release software to decrease the processing power requirement. In the same figure, each gray dot represents the center of each test point and is derived by the TaT algorithms. As shown in Fig. 5.20, some objects are the probe sites with large circular conductive pads whose outer rims are joined with one another. Meanwhile, the ADD algorithms determine the presence of multiple objects using the two-step process: first, placing two circles side-by-side inside the object, and, then, replacing all pixels in the two circles with black pixels. Repeat the two steps until no additional circle fits into the object. This helps the sub-algorithms to determine the number of test points in the large circular object. The ADD algorithms work with most via holes with white–colored outer edges as the behavior to lighting of large conductive pad and that of the white-colored outer edges are identical, thereby resulting in the similar images.



(a) (b) (c)

Figure 5.20 Extraction results using multiple algorithms (a) original image (b) during image processing (c) object centers.

5.5 Conclusion

Two main topics are principally described in this chapter; (1) probe tip endpoint identification and (2) test point identification. In practice, pattern matching does not need to search over the image. A region of interest (ROI) can be roughly defined to cover the possible area. The search within the limited region will reduce the computing resource. At this step, the endpoint and test point are reported in pixel position which is the coordinates on the image plane. These two coordinates (x_e and x_g) are later mapped to robot coordinates using an estimation method together with coordinate transformation back to the robot coordinates. A complete transformation of the related coordinates will be described in the subsequent chapter.

CHAPTER 6
VISUAL CALIBRATION AND POSE CORRECTION

6.1 Introduction

This chapter describes visual calibration, pose estimation and pose correction thereby the proposed method [20]. Instead of solving the projection equations exactly, an estimation method is used to minimize the sum of errors. Also, typical calibration is discrete method. Hence, it needs to numerically estimate the target position from the existing calibration data.

After mechanical tuning, the most critical part of the implemented robot is rotational; this is the head of robot moving around the Z-axis as shown in Fig. 6.1. In the rotational calibration, the implemented Cartesian robot uses a Nano Tech stepper motor (L4118 model), which has been tested by the manufacturer to guarantee high accuracy.

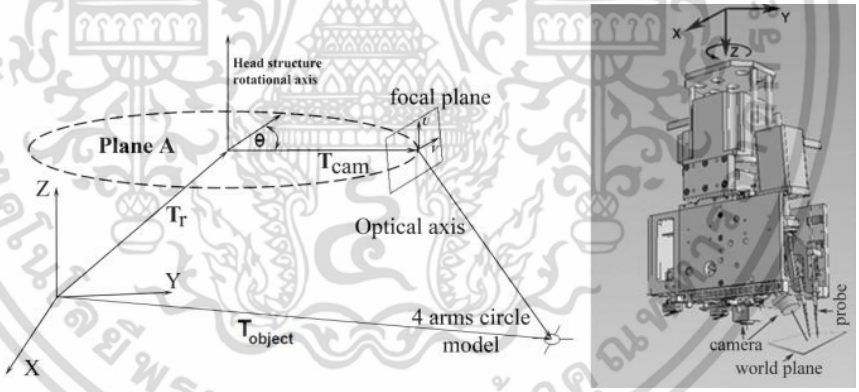


Figure 6.1 Robot coordinates system (left) and the head structure (right)

However, robot errors are inevitable with the addition of components to the hardware structure and/or of more wiring to the motor, which is widely practiced. This chaper describes a rotational calibration technique to calibrate the head structure rotation. The calibration with the proposed technique produces two results: the rotational linearity and the relationship of the angle and distance of the image coordinates to those of the world coordinates. The relationship coordinates are recorded in form of array. The data array can be used later for pose correction.

6.2 Visual Calibration

The calibration begins with the robot traveling from a position 1000 microns away from the stationing control point (i.e. target); and then, without rotating the head structure, adjusting its X-Y position by several small steps until the end-effector endpoint reaches the target. The steps are repeated after the rotation of the robot's head structure by one additional degree. At each degree, the robot moves the end-effector one pixel closer to the target to avoid an infinite loop of robot-image servoing. The relationship between the image coordinates and world coordinates is also obtained during the robot's discrete traveling since the coordinates of the end-effector from the TEDA algorithm and those of the robot from the robot controller are known. The relationship is in relative distances and angles, which subsequently serve as the inputs for the pose approximation when correcting errors in a real servoing task.

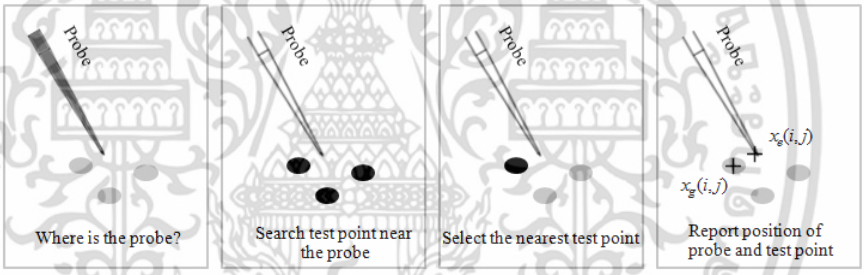


Figure 6.2 End-effector placement error: (a) find probe's endpoint (b) test point detection (c) select test point (d) report coordinates of probe and test point

Fig. 6.2 illustrates the reason why the relationship of the image coordinates and world coordinates must be known prior to pose correction. Fig. 6.2(a) illustrates the original image of the area of interest. Firstly, the probe endpoint is identified by using TEDA algorithm. Fig. 6.2(b) illustrates the detection result (i.e., test point) using the TDEA algorithm. Fig. 6.2(c) illustrates the final result replete with error information (x_e and x_g). The error information is in the unit of pixel, as shown in Fig. 6.2(d). Nevertheless, the information on the angles and distances of the world coordinates in micron for the pose correction is not available to the robot. Therefore, the relative angles and distances of the two coordinates are needed.

Prior research works on visual calibration used the knowledge of geometry to determine the relationship between the object and image coordinates. In addition, the relationship between the object and image coordinates could be determined by the camera calibration methods for a wide area, an example of the methods is Qiaoyu's method [25]. In Qiaoyu's work, prior to calibration the control points were created for the camera's wide view to enable the algorithm to extract all the control points and estimate the distances between pairs of all the control points in the area of interest. This current work, in contrast, utilizes this robot. Since the robot is faced with few distance errors and almost all the errors surround the end-effector, it thus requires a small area for the calibration.

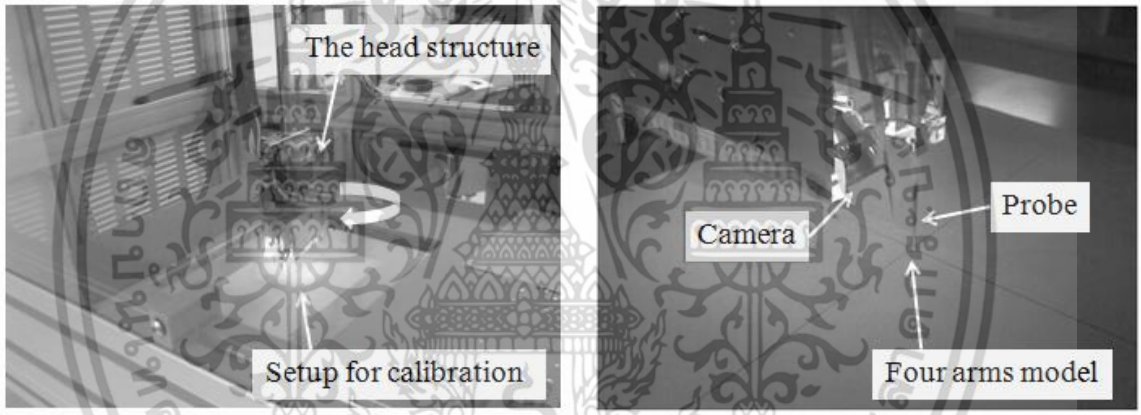


Figure 6.3 Mechanical setup for rotational calibration

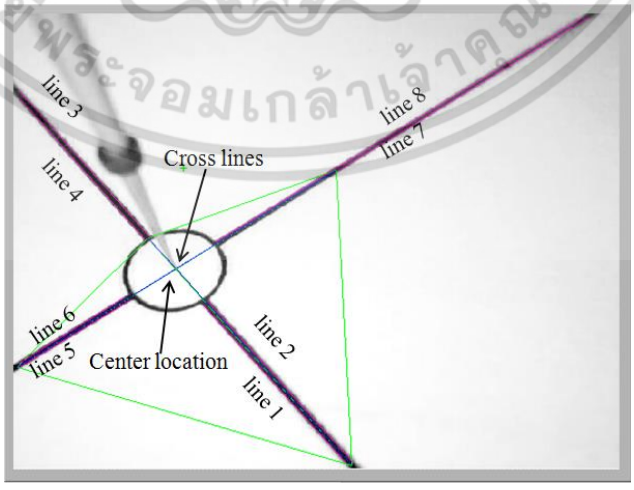


Figure 6.4 Example extraction results of four arms detection.

In addition, in the calibration extraction of the single control point must overcome obstruction to the camera by the end-effector. Therefore, a circular model connected to four arms was designed to enhance extraction at all rotational angles. Fig. 6.3 shows mechanical setup for the rotational calibration. Fig. 6.4 shows example extraction results of four arms detection.

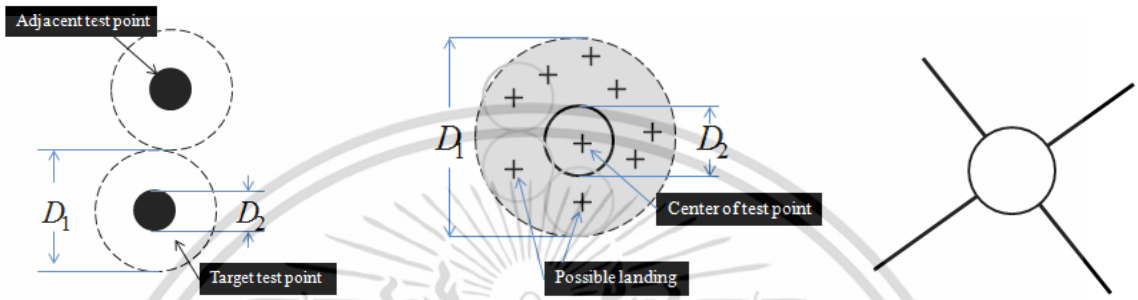


Figure 6.5 (a) Calibration area; (b) possible landing locations; (c) circular model for image extraction

The end-effectors, although from the same four-arm model, are different for different angles of view. As such, the camera calibration without the end-effector is not recommended. Appendix C describes more detail on the control point extraction. The circle control point was proposed by Ankur [25], who reported that the circle center localization had an error of 0.23 to 1.7 pixels whereas the center localization provided by NI has an error of 3 to 15 pixels, which are too large for this robot. The proposed center localization in this current research work yields a smaller error (0 - 1 pixels) in comparison with that of NI localization.

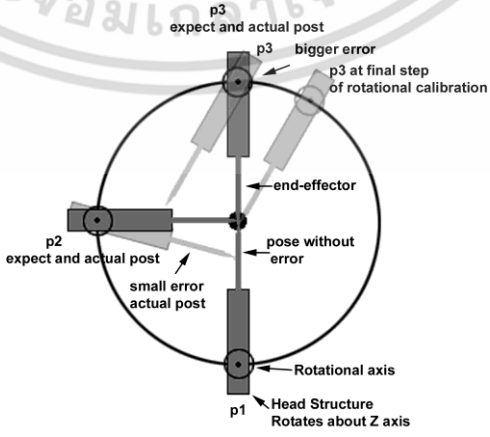


Figure 6.6 Top view of rotational calibration

The maximum calibration radius is set at $D_1/2$, Fig. 6.5a, where D_1 is the diameter of total hardware error. In practice, the diameter can be lengthened to include the error area, and the widened diameter can be later shortened in view of the maximum error obtained from the actual operation. Hence, the 1,000-micron radius is initially specified for calibration. The center position will be extracted from the four-arm model, Fig. 6.5c, and used as the control point. The robot continues traveling in small steps from a point in the calibration area to the control point according to the step sizes presented in Table 6.1.

Table 6.1 Robot step sizes

d_{im} (pixels)	Real world coordinates at rotation θ_i	
	Step size d_w (microns)	New coordinate
$10 \leq d$	300	$[x_o - d_w \cos\theta_i, y_o - d_w \sin\theta_i]$
$6 \leq d < 10$	200	
$4 \leq d < 6$	100	
$3 \leq d < 4$	50	
$d < 3$	30	

The robot then records the actuator position with the help of the camera when the end-effector has reached the control point at the first rotational degree of calibration. Under the assumption that X-Y has been well calibrated, plotting of the final X-Y coordinates of every degree produces a perfectly round circle as shown in Fig. 6.6 in which the plot circularity confirms the rotation linearity. If errors exist, the plotting of the X-Y coordinates will not produce a perfectly round circle.

Through the rotational calibration, both the rotation linearity and the relationship between the image and robot coordinates are known. The transformation of the image coordinates into the robot coordinates is by Eq. 6.1.

$$ {}^{W}\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = {}^{W}\begin{bmatrix} X_i \\ Y_i \end{bmatrix} + {}^{W}_{C}T_{\theta}\begin{bmatrix} x_{image} \\ y_{image} \end{bmatrix}, \tag{6.1} $$

where ${}^{W}_{C}T_{\theta}$ is the linear transformation of the pixel distance of each angle to the robot coordinates. Since the robot travels in a discrete manner, the interpolation technique is

employed to create the pseudo-continuous traveling. As presented in Table 6.1, the angles and distances of the discrete movement of the robot respectively increase in the one-degree step and by micron. Although the best resolution from the stepper motor is obtained at 0.05625 degree per step, the one-degree resolution per step is selected to reduce the calibration time. In the proposed calibration, the lens distortion and camera parameters can be ignored. Hence, this calibration works with most current high resolution cameras and future cameras with more advanced specifications.

To localize the target control point, another algorithm is proposed to extract the control point from the four-arm circular model. This model is a printout from a laser printer which is then placed on the workspace (Fig. 6.3). Even though each individual arm is one single thin line, the line extraction produces an appearance of two parallel lines for each arm. The upper and lower middle points that form a line between the two parallel lines of each arm are then computed. The intersection angles of any two arms are between 72 and 95 degrees due to the camera perspective. Other lines with an angle outside 72 and 95 degrees are removed and regarded as noise or unwanted lines. In the worst-case scenario in which the end-effector obstructs an arm from the camera's view, the intersection point can be calculated from only two lines with an angle between both lines of 72 to 95 degrees. Since at certain angles the probe tip could obstruct some parts of any two arms, the proposed algorithm thus has to be robust in calculating the central control point. With the proposed localization method, the central control point can be identified at any rotational angle as the robot rotates around the control point.

The rotational calibration steps are outlined below:

1) Servo the robot arm to the computed world position *(X,Y,Z)*.
2) Fix *X* and *Y*, and rotate θ in plane A to θ_{i+1}.
3) Acquire an image and extract the locations of the end-effector and the control point from the acquired image.
4) Servo the end-effector to a new location $[X_{i+1}, Y_{i+1}, Z]$, where $[X_{i+1}, Y_{i+1}, Z] = [X_i - d_w \cos\theta, Y_i - d_w \sin\theta, Z]$, where d_w is the robot step size in Table 6.1.
5) Repeat Steps 3-4 until convergence, (see the sample moves in Fig 6.7), and record the last position of the robot *(X,Y,Z)*.

6) Record in each moving step the relationship between the image coordinates and the known world coordinates (*X*, *Y* and *Z*) obtained from the robot controller.
7) Repeat Steps 1-6 until completion of rotational degree calibration. Note that possible of robot's rotation are between 0-100 and 141-360 degrees due to wiring limitation.

1st move 2nd move 3rd move 4th move 5th move 6th move

Figure 6.7 Sample moves of the end-effector toward the control point

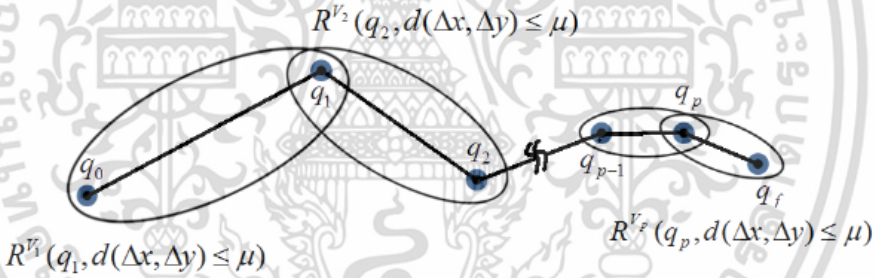


Figure 6.8 The moving steps from q_0 toward q_f

The moving steps are depicted in Fig. 6.8, where q_0 is the start point (q_{start}) and q_f is the final point (q_{goal}) in the space Z^2. In reality, the trajectory between q_0 and q_f is not a corridor path due to several unknown real distances and angles (i.e. γ) in the world coordinates [28]. Therefore, in the proposed calibration method, the target, q_f, is approached in a small-step fashion in the direction of ω (i.e., the angle seen in the image which guiding each step to move closer to the target) with Δx_{i+1} and Δy_{i+1} distances as shown in Table 6.1. The moving configuration can be described by $q_i = (\hat{a}_i d_w) = (x_i, y_i, \gamma_i)$ where $i \in \{1, \ldots, n\}$. Without slipping distance in each move, the next move is written as Δx_{i+1} and Δy_{i+1}.

$$\begin{bmatrix}\Delta x_{i+1}\\ \Delta y_{i+1}\end{bmatrix}=\begin{bmatrix}\cos\gamma_i & \sin\gamma_i\\ -\sin\gamma_i & \cos\gamma_i\end{bmatrix}\begin{bmatrix}d_{(x_e-x_g)}\\ d_{(y_e-y_g)}\end{bmatrix}, \tag{6.2}$$

q_0 and q_f can be written in a function of *X*, *Y* coordinates together with the real angles of $q_{start}=(x_{e,0},y_{e,0},\gamma_{e,0})$ and $q_{goal}=(x_{g,0},y_{g,0},\gamma_{g,0})$. Mapping of $\Delta\gamma\mapsto\Delta\omega$ is now known. The robot stops moving at the final position, *p*, when $\| q_{p+1}-q_f \|\le 1$ is reached. The relative mapping between the image and world coordinates can be written as:

$$(d_{im},\omega)\in Z^2\mapsto(d_w,\gamma)\in R^3, \tag{6.3}$$

The proposed calibration is also applicable to Z-height calibration because the 12-micron-pixel pitch offers sufficient resolution in differentiating the Z-height distance while no investment in the high resolution laser for Z distance measurement is required. Therefore, the TEDA algorithm can be used to test the planar of a metal base plate. The Z-height calibration starts with the robot traveling in a random fashion over the workspace without rotation of the head structure. The length of Z has been predetermined. At different random positions, images are acquired and fed into the TEDA algorithm to derive x_e. All derived are evaluated to see whether they repeat the same value. The repetition indicates that the metal base plate is planar.

6.3 Pose Estimation

The placement error of the robot's pose (position and orientation) can be determined by the distance and angle errors. Therefore, both errors can be used for the pose estimation. The rotational calibration in the above section records the relationship between the image and world coordinates with respect to distances and angles. The relative calibration data between the image and robot coordinates are then recorded. By ignoring the fixed length Z, the recorded data can be expressed with the following formulas:

$$f_{array}: d_{im,i}\angle\omega_i\mapsto d_{w,i}\angle\gamma_i : 0\le\gamma\le 2\pi, \tag{6.4}$$

$$d_{im,i}\angle\omega_i=\left|x_{e,i}(x,y)-x_{g,i}(x,y)\right|\angle\arctan(x_{e,i}(x,y)/x_{g,i}(x,y)), \tag{6.5}$$

$$d_{w,i}\angle\gamma_i = \left|d_{w,i}(x,y) - d_{w,0}(x,y)\right| \angle \arctan(d_{w,i}(x,y) / d_{w,0}(x,y)), \tag{6.6}$$

where the data array, f_{array}, consists of the relative mapping elements. One mapping element comprises the error distance, $d_{im,i}$, and angle, ω_i on the image, which are mapped to the error distance, $d_{w,i}$, and angle , γ_i, on the robot coordinates. The distance and angle can be calculated using the Euclidean angular distance. Since the control point is the stationing point, S (or x_g), the final moving step, p, thus represents the end-effector coordinates, P^e. Therefore, the kinematic error function can be expressed in base coordinates as

$$E_{pp}(x_e; S, P^e) = x_e(P^e) - S, \tag{6.7}$$

Total error, e, of each individual calibration can be calculated by Eq. 6.8, where n starts from 0 to 360 degrees. The 1-step degree is used for calibration.

$$\| e \|^2 = e_0^2 + e_1^2 + ,..., + e_{n-1}^2 + e_n^2, \tag{6.8}$$

Because of the hardware rotational limits of robot's rotation, the angles between 101-140 degrees are skipped. The robot total error can be more specifically explained by Eq. 6.9.

$$e = \sqrt{(\sum_{i=0}^{100} | x_{R(X,Y)} - x_{r(X,Y),i} | + \sum_{i=141}^{360} | x_{R(X,Y)} - x_{r(X,Y),i} |)}, \tag{6.9}$$

The first distance, $d_{w,0}$, from the actuator's controller at the first calibration degree is known once the end-effector has reached the target. The coordinates of every rotation, $d_{w,i}$, can be determined. The relationship between the image and robot coordinates at 174 degrees is shown in Fig. 6.9.

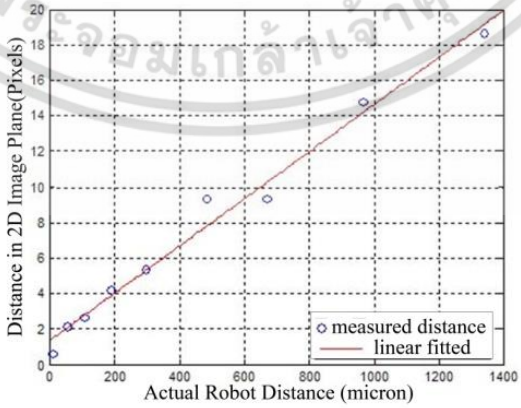


Figure 6.9. Relationship of the distance on image versus the robot distance at 174 degrees

The proposed centralization method gives the resolution of less than one pixel. To reduce calibration time, it is however permissible to set the error between one and up to two pixels. With the hardware issue solved, the robot produces more precise results, as shown in Fig. 6.11(b). In the same figure, 99.94% of 251 trials landed near the control point within the error length of 200 microns at the first jump. In this experiment, the robot figuratively opens its eye to confirm the accuracy. In the real servoing operation, even near 100% accuracy still requires the visual servo since errors from other sources could degrade the robot accuracy by approximately 5%, e.g. the fiducial marking errors, drill hole errors, and unavoidable mechanical errors. Thus, the visual servo is typically called upon at about 5% of the time. Fig. 6.10 visualizes circularity which impile linearlity of rotation of the head structure.

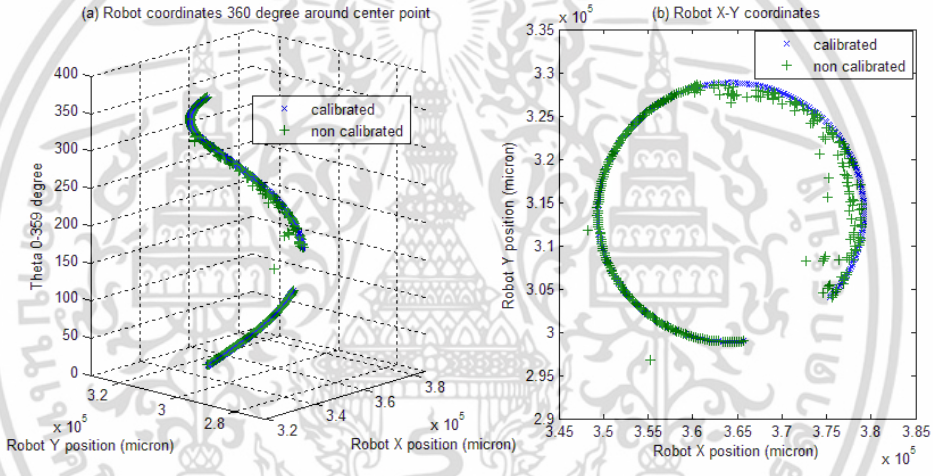


Figure 6.10 Rotational calibration around Z axis: (a) Linearity of rotation around Z axis (b) plot of X-Y coordinates around Z axis

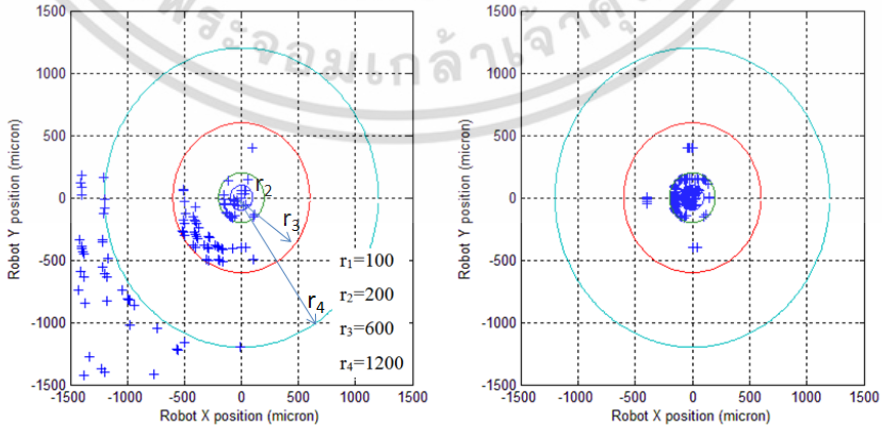


Figure 6.11 Experimental results: (a) non-calibrated and (b) calibrated landing

Figs. 6.12(a) to 6.12(i) show the detection results of the probe tip and test points in different areas. The centers of blue squares are the locations of the end-effector and those of green squares inside the green circles are the locations of test points. The locations of the end-effector, x_e, and the test points, x_g, are listed in Table 6.2. In the same table, the error information is mapped to the robot's angular distance as the final output data for pose estimation. The reverse of the pose estimation in Table 6.2 will be fed into the software program to instruct the robot to re-attempt placement of the probe tip.

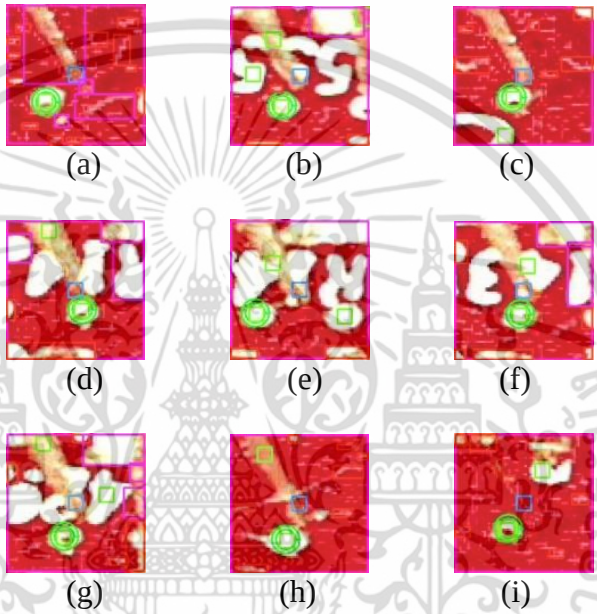


Figure 6.12 Detection of the tip endpoint and test points at different locations

Table 6.2. Detection Coordinates and Robot Estimation

Picture	Image coordinates		Estimation
	x_e ,x_g	$d_{im} \angle \omega$	$d_w \angle \gamma$
a	(426,624), (412,638)	19.8 ∠ -0.78	6233 ∠ 0.79
b	(442,621), (434,636)	17.00 ∠ -0.49	5577 ∠ 1.08
c	(421,613), (416,626)	13.92 ∠ -0.36	4566 ∠ 1.02
d	(424,628), (428,638)	10.77 ∠ 0.38	3533 ∠ 1.95
e	(429,626), (408,638)	24.18 ∠ -1.05	7874 ∠ 0.52
f	(435,616), (434,628)	12.04 ∠ -0.08	3950 ∠ 1.49
g	(435,607), (430,624)	17.72 ∠ -0.28	5813 ∠ 1.28
h	(433,630), (428,642)	13.00 ∠ -0.39	4265 ∠ 1.17
i	(430,635), (434,640)	6.04 ∠ -0.67	2099 ∠ 0.89

6.4 Pose Correction

Pose correction is last step of the visual servo. Since the camera is mounted in the eye-in-hand configuration, it needs to translate the camera position on the robot head structure also. However, in this work since the head structure is a rigid body and the controller requires only relative distance and angle of the pose in order to perform the pose correction.

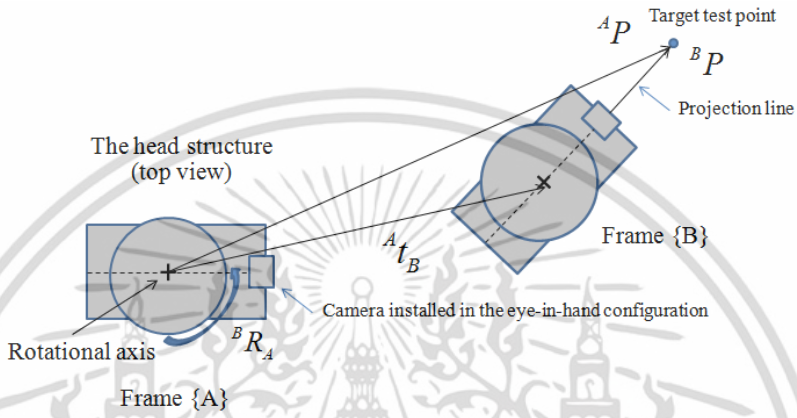


Figure 6.13 Illustration of transformation of the robot's head structure

With reference to Fig. 6.13, the head structure of robot is originally located at ${}^{0}P(X,Y,Z)$, after the robot has completely stopped at the desired position ${}^{1}P(X,Y,Z)$ in order to perform fault insertion signal on to a test point on the PCB under test. The transformation can be expressed as:

$$ {}^{B}P = {}^{B}H_{A}\,{}^{A}P = \begin{pmatrix} {}^{B}R_{A} & {}^{B}t_{A} \\ 0 & 1 \end{pmatrix} {}^{A}P, \tag{6.10} $$

and

$$ {}^{A}P = {}^{A}H_{B}\,{}^{B}P = \left({}^{A}H_{B}\right)^{-1}\,{}^{B}P = \begin{pmatrix} {}^{A}R_{B} & {}^{A}t_{B} \\ 0 & 1 \end{pmatrix} {}^{B}P, \tag{6.11} $$

where ${}^{A}P$ and ${}^{B}P$ are the representation of P (i.e., test point) in frame {A} and {B} respectively. ${}^{A}t_{B}$ is translation in the {A} to {B}. H represents a single transformation homogenous matrix. In this application, the terms rotation, R, can be reduced into rotation around the Z axis, 45^{o}, for example. In practice, R and t will be optimized in order to get available path to servo the head structure to the target area due to obstruction and angle limitation. The translation is [10,5,0], for example. Below is an example of the coordinate transformations which also describes the pose correction.

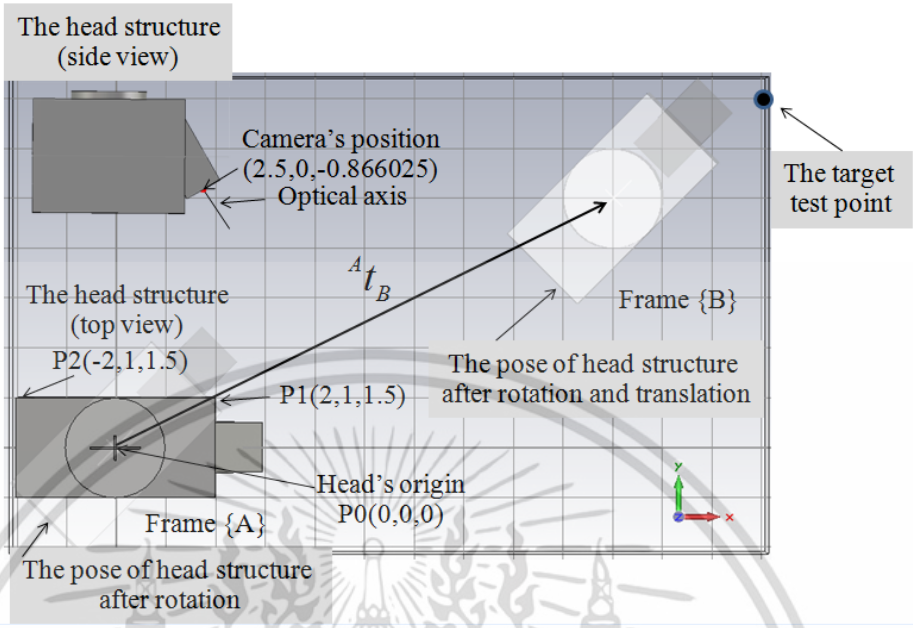


Figure 6.14 Example of reference coordinates on the workspace

Fig. 6.14 shows example reference coordinates. In this example, translation of the camera and the probe are ignored due to the fact the feedback position will be retrieved from the mapping array of the pose estimation thereby the feedback position is already in the real world coordinates. Hence, the correction information in terms of world position for the robot's controller can be converted respectively as follow:

$$
{}^{B}P = \begin{pmatrix} {}^{B}R_{A} & {}^{B}t_{A} \\ 0 & 1 \end{pmatrix} {}^{A}P = \begin{pmatrix} \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 1 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 10 \\ 5 \\ 0 \end{pmatrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{pmatrix} \begin{pmatrix} 0 & 2 & -2 & 2.5 \\ 0 & 1 & 1 & 0 \\ 0 & 1.5 & 1.5 & -0.866 \\ 1 & 1 & 1 & 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} 0.7071 & -0.7071 & 0 & 10 \\ 0.7071 & 0.7071 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 2 & -2 & 2.5 \\ 0 & 1 & 1 & 0 \\ 0 & 1.5 & 1.5 & -0.866 \\ 1 & 1 & 1 & 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} 10 & 10.7071 & 7.8787 & 11.7678 \\ 5 & 7.1213 & 4.2929 & 6.7678 \\ 0 & 1.5 & 1.5 & -0.866 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{6.12}
$$

The latest outcome of Eq. 6.12 shows coordinates of the reference points on the head structure in frame {B}, [11.76,6.76,-0.866] is position of camera after rotation and translation, while a test point is positioned at [13,7,-0.8660] to intentionally produce an error for further correction. In the view of camera, first, the error is detected in image coordinates (i.e., x_e and x_g). In this example, for perfect pose correction, the error in world coordinates should be mapped to ${}^{map}P$=[-1.2322,-0.2322,0,0]. However, the mapping position must be transformed back to the coordinate frame {A}, ${}^{A2}P$, as follows:

$$ {}^{A2}P = \left({}^{A}H_{B} \right)^{-1} {}^{map}P = \begin{pmatrix} 0.7071 & -0.7071 & 0 & 10 \\ 0.7071 & 0.7071 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} -1.2322 \\ -0.2322 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1.0355 \\ -0.7071 \\ 0 \\ 0 \end{pmatrix} \tag{6.13} $$

Now, outcome of Eq. 6.13 can be fed as feedback information, ${}^{A2}P$, to the robot to recalculate the new position, ${}^{B2}P$, for the next move by using Eq. 6.12 together with the feedback information.

$$ {}^{B2}P = \begin{pmatrix} {}^{B}R_{A} & {}^{B}t_{A} \\ 0 & 1 \end{pmatrix} \left({}^{A}P - {}^{A2}P \right) $$

$$ = \begin{pmatrix} 0.7071 & -0.7071 & 0 & 10 \\ 0.7071 & 0.7071 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 0 & 2 & -2 & 2.5 \\ 0 & 1 & 1 & 0 \\ 0 & 1.5 & 1.5 & -0.866 \\ 1 & 1 & 1 & 1 \end{pmatrix} - \begin{pmatrix} -1.0355 & -1.0355 & -1.0355 & -1.0355 \\ 0.7071 & 0.7071 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right) $$

$$ = \begin{pmatrix} 11.2322 & 11.9393 & 9.1109 & 13 \\ 5.2355 & 7.3535 & 4.5251 & 7 \\ 0 & 1.5 & 1.5 & -0.866 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{6.14} $$

The reference coordinates of the last column in the Eq. 6.14 now touch the target test point. An incorrect placement of the end-effector can be corrected using this calculation. The correction data can be calculated using the bilinear interpolation technique from the angles and distances in the mapping array to get the ${}^{map}P$ of each angle. In the actual operation, the total error can be thresholded to alert the operator to undertake the preventive maintenance also.

6.5 Conclusion

This chapter described the proposed visual calibration, pose estimation and pose correction. The proposed rotational calibration can be achievable by using the two detection algorithms; probe tip endpoint detection and control point detection. Through the rotational calibration, both the rotation linearity and the relationship between the image and robot coordinates are known. The known mapping data is recorded for simple pose estimation and pose correction. The four-arm model can be easily printed out by using a typical laser printer to place on the workspace to enable the robot's visibility. Low cost, practical and precise calibration can be obtained through the proposed calibration. Moreover, pose correction can be made by retrieving the error information out of the relationship between the image and world coordinates which are obtained during calibration.

CHAPTER 7
CONCLUSION AND REMARK FOR FUTURE STUDIES

7.1 Conclusion

This thesis details the background knowledge of PBVS toward design and implementation of the proposed PBVS. The proposed visual servo system is categorized as a Position-Based in eye-in-hand configuration. The PBVS is designed and implemented for pose correction of the end-effector placement error of a Cartesian robotics. The thesis includes a novel detection of the probe tip and test points. The proposed PBVS is simple and very practical for handling the operational factors. The details of the robot calibration are also provided and the rotational calibration requires approximately twenty-one minutes. As the mechanical and human errors are inevitable even with a well-calibrated machine, the complete visual servo is required for the introduced robot at the run time mode to increase the test yield. The test yield can be improved by pose correction when errors occur. The correction can be made in less than three seconds per one correction loop without complex computation as the moving relationship has been made available for data interpolation to calculate the pose estimation.

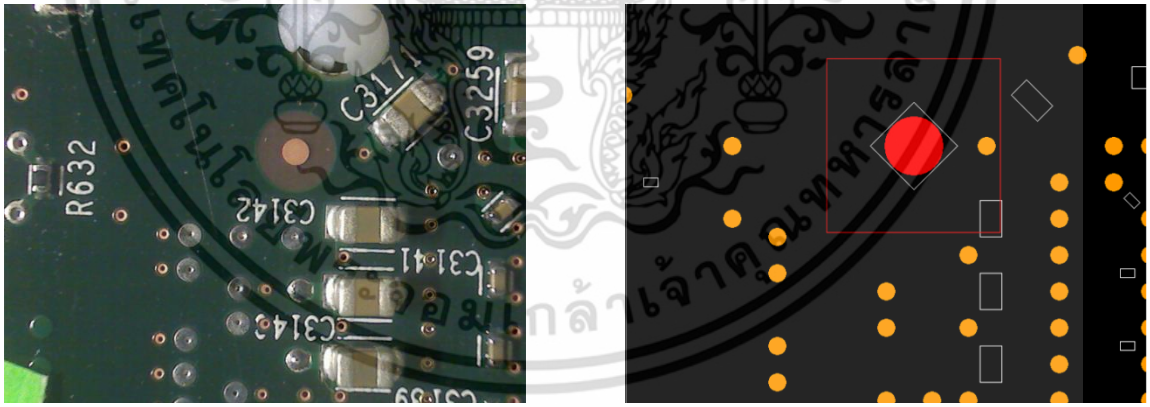
The Nearest Neighbor (NN) algorithms [29] could be applied in searching for a test point, and the TDEA algorithm stops analyzing other objects once the nearest test point is identified. However, NN is unsuitable for the Cartesian robot as multiple objects can be tangent to each other at the outer edges. Thus, the center of the object's ROI cannot instantly identify its nearest neighbor because the object requires analysis beforehand to recognize the entire object.

Speed is another cause of concern since the addition of algorithms increases the execution time. This work, however, made an investment in a high performance server as it accounts for a fraction of the total costs of the robot project. Thus, the issue of speed could be left out. In addition, the application is a low-speed and is normally used for product evaluation during the product design phase prior to mass production. Therefore,

test time is not key to the design of the software. The total execution time required is minimal at approximately one second each for TEDA and TDEA. Also, the total test time per one call to visual servo is less than three seconds, the length of time which includes image acquisition using the existing test equipment. In fact, in comparison with test time, reliability and accuracy are valued more highly by manufacturers and customers.

7.2 Remark for Future Works

Current selection method of test point is to simply move to the closed test point next to the end-effector. The correction could false moving to wrong test point if more than one test points with the same distance are detected. Future research could introduce a better test point selection method to address the problem of wrongly selecting the test points. The constellation of test points in the memory facilitates the searching by the algorithm and thereby addresses the problem of wrong selection. Moreover, to solve the wrong selection problem, the constellation of test points near the target area directly extracted from CAD is compared against the constellation of test points extracted from the image, Fig. 7.1.



(a) (b)

Figure 7.1 (a) Example PCB image (b) constellation of test points extracted from CAD of the example PCB

The designers and developers should bear in mind that there are several factors involved in the implementation of all parts as a system, such as the poor reflection caused by oxide at the test point of an old PCB. The designers and developers should also

include the workflow to handle the system constrains. The visual servo should be able to classify which object is a test point and which is not, and it should return a false value to the software program if the classification cannot be made. In addition, a safety sensor and a limit switch should be installed to protect the hardware and the unit under test in the event of software and hardware malfunctions.

REFERENCES

[1] S. Hutchinson, G.D. Hager, and P.I. Corke, "A Tutorial on Visual Servo Control," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 651-670, Oct. 1996.

[2] B.H. Yoshimi and P.K. Allen, "Alignment Using an Uncalibrated Camera System," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 516-521, Aug. 1995.

[3] S. Moon, S.G. Lee, and K.H. Park, "Recent Progress of Robotic Vocabulary Standardization Efforts in ISO," *SICE Annual Conference 2010*, pp. 266-268, Aug. 2010.

[4] S.E. Hodges, "Looking for a Cheaper Robot: Visual Feedback for Automated PCB Manufacture," *Ph.D. dissertation, University of Cambridge*., Jan. 1996.

[5] F.D. Turek, "Machine Vision Fundamentals, How to Make Robots See," *NASA Tech Briefs 35*, Nov. 2011.

[6] M. Vincze, and G.D. Hager, "Robust Vision for Vision-Based Control of Motion," *Wiley-IEEE Press*, Feb. 2000.

[7] A.C. Sanderson, and L.E. Weiss, "Image-based visual servo control using relational graph error signals," *Proc. IEEE*, pp.1074-1077, 1980.

[8] J. Feddema and O. Mitchell, 'Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. Robot. Automat*, vol. 5, pp. 691-700, Oct. 1989.

[9] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," *The confluence of vision and control*, vol. 237, pp.66-78, 1998.

[10] F. Chaumette and S. Hutchinson, "Visual servo control. I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82-90, Dec. 2006.

[11] A.K. Ray, M. Agarwal, and L. Behera, "Kinematic Control of Robot Manipulators using Visual Feedback," *Proc. of the 2006 IEEE International Symposium on Intelligent Control*, pp 3076-3081, Oct. 2006.

[12] E.R. Davies, "Machine Vision: Theory Algorithms, Practicalities," *2nd Edition Academic Press*.

[13] R.C. Gonzalez, R.E. Woods, and S.L. Eddins, “Digital image Processing Using Matlab,” *New Jersey, Prentice Hall*, 2003.

[14] G.X. Ritter, and J.N. Wilson, “Handbook of Computer Vision Algorithms in Image Algebra,” *2nd Edition CRC Press*, 1996.

[15] “National Instruments Corporation,” *NI Vision Concepts Manual*, Nov. 2005.

[16] “National Instruments Corporation,” NI *Vision for LabVIEW User Manual*, Nov. 2005.

[17] N. Otsu, “A Threshold Selection Method from Gay-Level Histograms,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, Jan. 1979.

[18] T. Banlue, P. Sooraksa and S. Noppanakeepong, “Non-linear Visual Servo System for Printed-Circuit Board Fault Injection Test,” The Fifth Asia-Pacific Workshop on Chaos Control and Synchronization *(APWCCS)*, Kunming, China, Oct. 2010.

[19] T. Banlue, P. Sooraksa, and S. Nopppanakeepong, “A Practical Visual Servo Calibration of End-Effector Pose Estimation in Industrial Manufacturing,” *International Conference on Engineering, Applied Sciences, and Technology*, Nov. 2012.

[20] T. Banlue, P. Sooraksa and S. Noppanakeepong, “A Practical Position-Based Visual Servo Design and Implementation for Automated Fault Insertion Test,” International Journal of Control, Automation, and Systems (IJCAS), *vol. 12, no. 5, Oct. 2014.*

[21] S. Yi, R.M. Haralick, and L.G.Shapiro, “Automatic Sensor and Light Source Positioning for Machine Vision,” *Pattern Recognition, 1990. Proc. of 10th International Conference*, vol. 1, pp. 55-59, 1990.

[22] M. Sezgin, and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging*, vol. 13, pp.146-165, Jan. 2004.

[23] X. Gao, C. Zhao, G. Chang, and Z. Tan, “Design and Implementation of Drill Hole of PCB Detection System,” *IEEE 9th International Conference on Hybrid Intelligent Systems*, vol. 2, pp. 462-466, Sep. 2009.

[24] P.R. Giordano, A. Stemmer, K. Arbter and A. Albu-Schaffer, “Robotic Assembly of Complex Planar Parts: An Experimental Evaluation,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 3775-3782, Sep. 2008.

[25] Q. Xu, D. Ye, R. Che, and Y. Huang, “Accurate Camera Calibration with New Minimizing Function,” *IEEE International Conference on Robotics and Biomimetics*, pp. 779-784, Dec. 2006.

[26] A. Datta, J.S. Kim, and T. Kanade, “Accurate Camera Calibration using Iterative Refinement of Control Points,” *2009 IEEE 12th International Conference on Computer Vision Workshops*, ICCV Workshops, pp. 1201-1208, 2009.

[27] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.

[28] A. Becker and T. Bretl, “Approximate Steering of a Unicycle Under Bounded Model Perturbation Using Ensemble Control,” *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 580-591, 2012.

[29] G. Shakhnarovich, T. Darrell, and P. Indyk, “Nearest-Neighbor Methods in Learning and Vision: Theory and Practice,” *MIT Press*, 2006.

[30] J. Heikkila, “Geometric Camera Calibration Using Circular Control Points,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066-1077, Oct. 2000.

[31] H.F. Ng, “Automatic thresholding for defect detection,” *Proc. of the 3rd International Conference on Image and Graphics*, pp. 532-534, 2004.

[32] L. Gracia, and C. Perez-Vidal, “A New Control Scheme for Visual Servoing,” *International Journal of Control, Automation, and Systems*, vol. 7, no. 5, pp.764-776, Oct. 2009.

[33] W. Yu, “Stability analysis of visual servoing with sliding-mode estimation and neural compensation,” *International Journal of Control, Automation, and Systems*, vol. 4, no. 5, pp. 545-558, Oct. 2006.

[34] Y. Zhao, W.F. Xie, and S. Liu, "Image-based Visual Servoing Using Improved Image Moments in 6-DOF Robot Systems," *International Journal of Control, Automation, and Systems*, vol. 11, no. 3, pp.586-596, Jun. 2013

[35] J. Ha “An Image Processing Algorithm for the Automatic Manipulation of Tie Rod,” *International Journal of Control, Automation, and Systems*, vol. 11, no. 5, pp.984-990, Oct. 2013.

APPENDICES

APPENDIX A

Jacobian Iterative Estimation

Jacobian matrix is the matrix of partial derivatives of that function. In numerical, Jacobian iterative method is an algorithm for estimating the solutions of a diagonal dominant system of linear equations. The estimation outcome will be approximated value. The value will be numerically iterated until it converges. For a continuous transformation from (x,y) to (u,v). Then, $x=x(u,v)$ and $y=y(u,v)$.

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y = \iint_{R'} f(x(u,v),y(u,v))\frac{\partial(x,y)}{\partial(u,v)}\mathrm{d}u\mathrm{d}v \tag{A.1}$$

$$\left[\frac{\partial(x,y)}{\partial(u,v)}\right] = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix} \tag{A.2}$$

where $\mathrm{F}:\Re^2 \to \Re^2$ is so-called 2D Jacobian thereby maps areas dxdy to areas dudv. 3D Jacobian can be also described as: $x=x(u,v,w)$, $y=y(u,v,w)$ and $z=z(u,v,w)$.

$$\iiint_V f(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z = \iiint_{V'} f(u,v,w)\frac{\partial(x,y)}{\partial(u,v)}\mathrm{d}u\mathrm{d}v\mathrm{d}w \tag{A.3}$$

$$\left[\frac{\partial(x,y,z)}{\partial(u,v,w)}\right] = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} & \frac{\partial z}{\partial w} \end{bmatrix} \tag{A.4}$$

Typical algorithm of a Jacobian iterative method to find x for $y=f(x)$, which minimizes error $E=|f(x)-y_0|^2$ can be outlined below:

1) Start with x_0 as guess for x,
2) Compute $y=f(x)$. Residual error becomes $dy=y-y_0$.
3) Construct Jacobian of f where $\left[\partial f / \partial x\right]$ and $dy=Jdx$.
4) Compute dx using pseudo inverse $dx=(J^TJ)^{-1}J^Tdy$.
5) Update estimation of $x=x+dx$.
6) Repeat step 2-5 until E can be acceptable or convergence.

Example of finding x in numerical iterative method for $\tilde{p} = KM_{ext}\,{}^{M}P = K\left({}^{C}R_W \quad {}^{C}t_W\right){}^{M}P$. where M_{ext} is unknown matrix and $y = \tilde{p}$. where K and ${}^{M}P$ are known intrinsic and model coordinates respectively. Recall Eq. 3.29

$$M_{ext} = \left({}^{C}R_W \quad {}^{C}t_w\right) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{pmatrix} \tag{A.5}$$

$$M_{ext} = \left({}^{C}R_W \quad {}^{C}t_w\right) = \left[\left[{}^{C}R_W\right]^{3\times3} \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}^{1\times3}\right]^{4\times3} = \left[\left[R_{\theta_Z} R_{\theta_Y} R_{\theta_X}\right]^{3\times3} \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}^{1\times3}\right]^{4\times3} \tag{A.6}$$

$$= \left[\left[\begin{pmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & -\sin\theta_X \\ 0 & \cos\theta_X & \cos\theta_X \end{pmatrix}\right]^{3\times3} \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}^{1\times3}\right]^{4\times3} \tag{A.7}$$

$$y=\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\left[\left[\begin{pmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & -\sin\theta_X \\ 0 & \cos\theta_X & \cos\theta_X \end{pmatrix}\right]^{3\times3}\begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}^{1\times3}\right]^{4\times3}$$
$$\begin{pmatrix} X_1 & X_2 & \ldots & X_6 \\ Y_1 & Y_2 & \ldots & Y_6 \\ Z_1 & Z_2 & \ldots & Z_6 \\ 1 & 1 & \ldots & 1 \end{pmatrix} \quad \text{(A.8)}$$

Then, compute $y=\tilde{p}=f(x)=f(\theta_X,\theta_Y,\theta_Z,t_X,t_Y,t_Z)$ with initial guess value (e.g. x=(1,1,1,1,1,1)). The Jacobian derivatives can be approximated as.

$$\frac{\partial f(x)}{\partial(x_i)}\approx\frac{f(x+\varepsilon\hat{u}_i)-f(x)}{\varepsilon} \quad \text{(A.9)}$$

$$J=\left[\frac{\partial f_i}{\partial x_j}\right]=\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_M} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_M} \\ . & & & . \\ . & & & . \\ . & & & . \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \ldots & \frac{\partial f_N}{\partial x_M} \end{pmatrix} \quad \text{(A.10)}$$

First column and last column of the Jacobian matrix can be example written in Eq. A.11 and A.12 respectively.

$$\left[\frac{\partial f_1}{\partial x_1}\ \ \frac{\partial f_2}{\partial x_1}\ \ \ldots\ \ \frac{\partial f_N}{\partial x_1}\right]^T=\frac{f\left(\left[\theta_X\ \ \theta_Y\ \ \theta_Z\ \ t_X\ \ t_Y\ \ t_Z\right]^T+\left[\varepsilon\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\right]^T\right)-f\left(\left[\theta_X\ \ \theta_Y\ \ \theta_Z\ \ t_X\ \ t_Y\ \ t_Z\right]^T\right)}{\varepsilon} \quad \text{(A.11)}$$

$$\left[\frac{\partial f_1}{\partial x_M}\ \ \frac{\partial f_2}{\partial x_M}\ \ \ldots\ \ \frac{\partial f_N}{\partial x_M}\right]^T=\frac{f\left(\left[\theta_X\ \ \theta_Y\ \ \theta_Z\ \ t_X\ \ t_Y\ \ t_Z\right]^T+\left[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ e\right]^T\right)-f\left(\left[\theta_X\ \ \theta_Y\ \ \theta_Z\ \ t_X\ \ t_Y\ \ t_Z\right]^T\right)}{\varepsilon} \quad \text{(A.12)}$$

APPENDIX B
Vision Software Tools

This work uses a family of National Instruments (NI) vision software. NI provides vision software tools, solutions and friendly custom framework for vision software development. Extensive vision applications use NI Vision for vision development in manufacturing. For vision system beginner, LabView and Vision Development are prerequisite software for installation and their evaluation version is free of charge.

The NI Vision software set includes:

1) NI Vision Assistant 8.6.0
2) NI LabView 8.6
3) NI Vision Development 8.6.0
4) NI Vision Builder 3.6
5) NI Vision Acquisition 2009

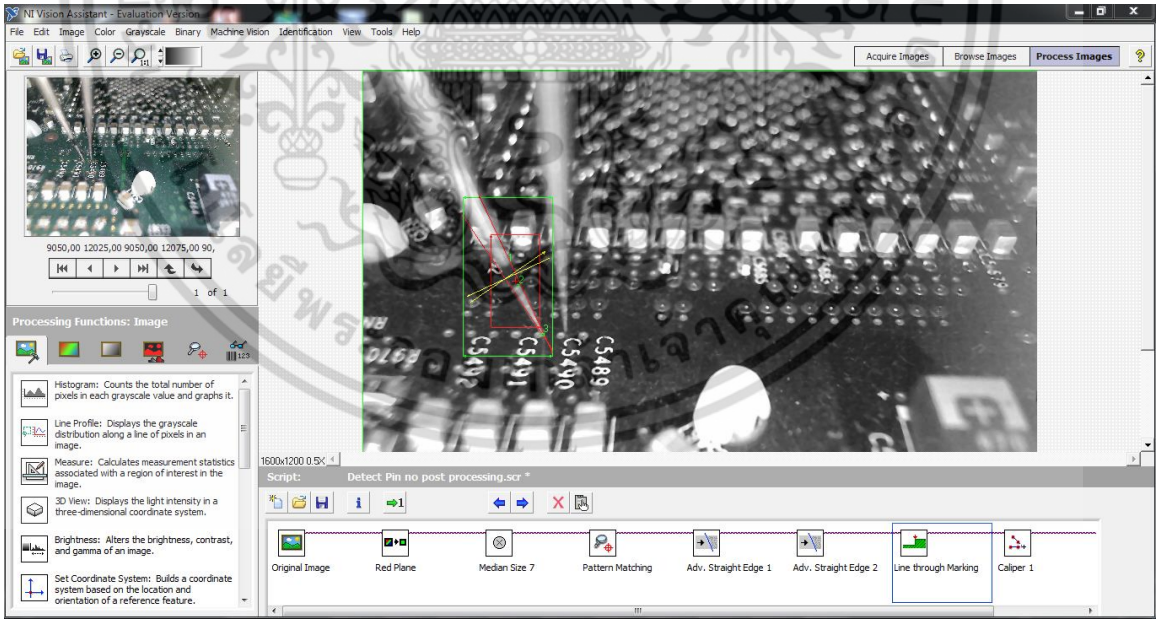


Figure B.1 Vision Assistant version 8.6.0

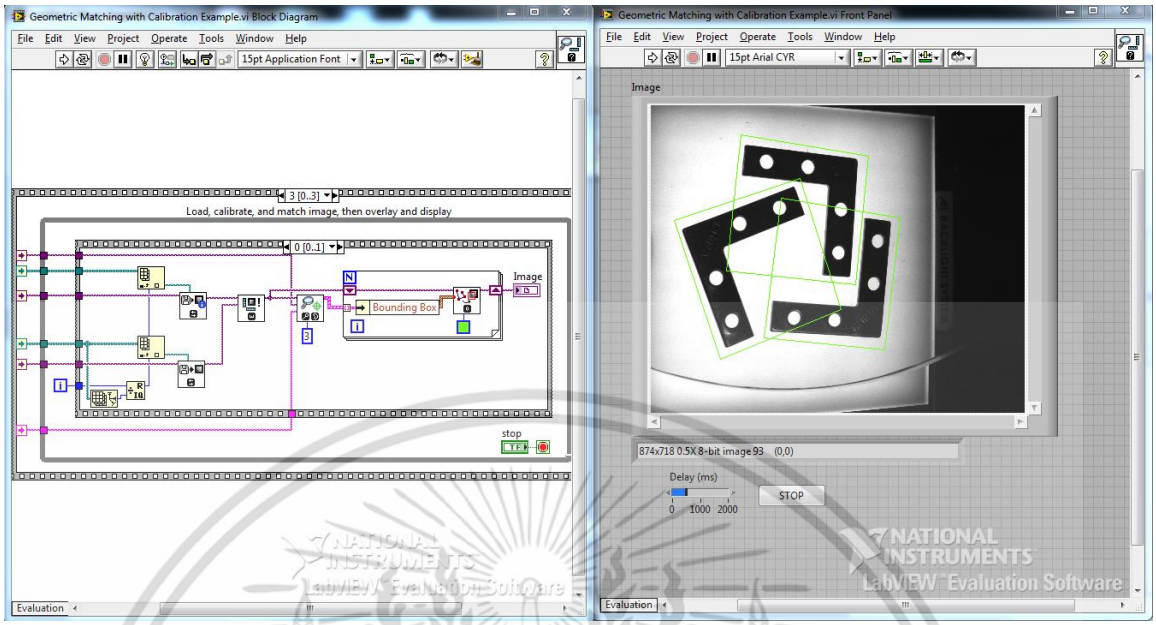


Figure B.2 LabView 8.6 included with Vision Development tools

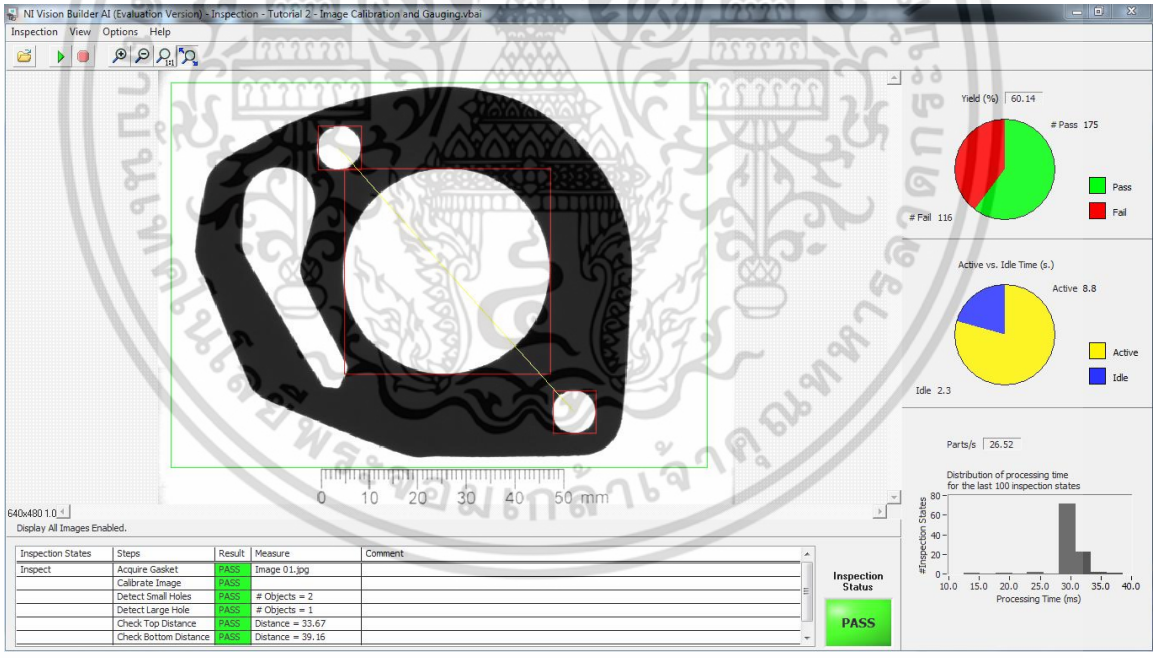
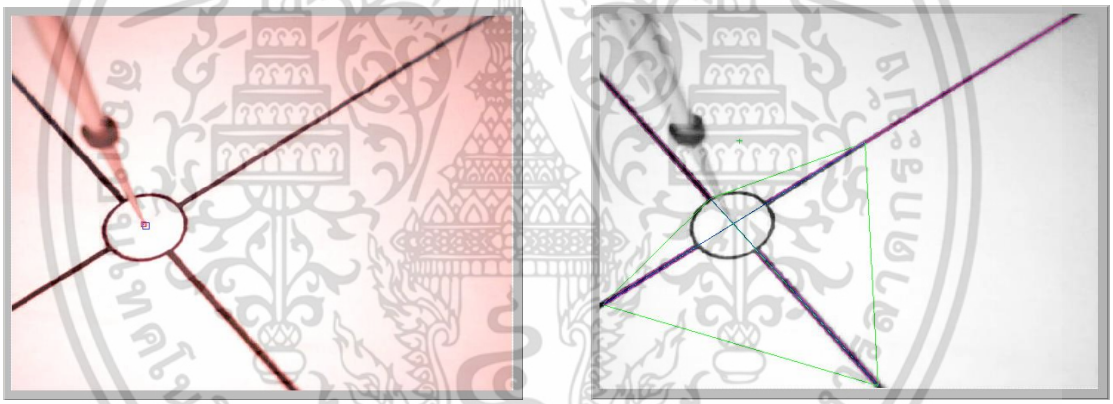


Figure B.3 Vision Builder

APPENDIX C
Control Point Identification

This sub-section describes identification algorithm of a control point. Control point is used for camera calibration and machine calibration. The calibration only requires observation of a planar pattern. Typical calibration uses camera to extract metric information from 2D images. Square control points [25], grid square [26], circular ring [25] are traditional control points. These control points are suitable for camera wide view. Fig. C.1 shows four-arm circular control point which is designed for higher accuracy positioning in narrower area.

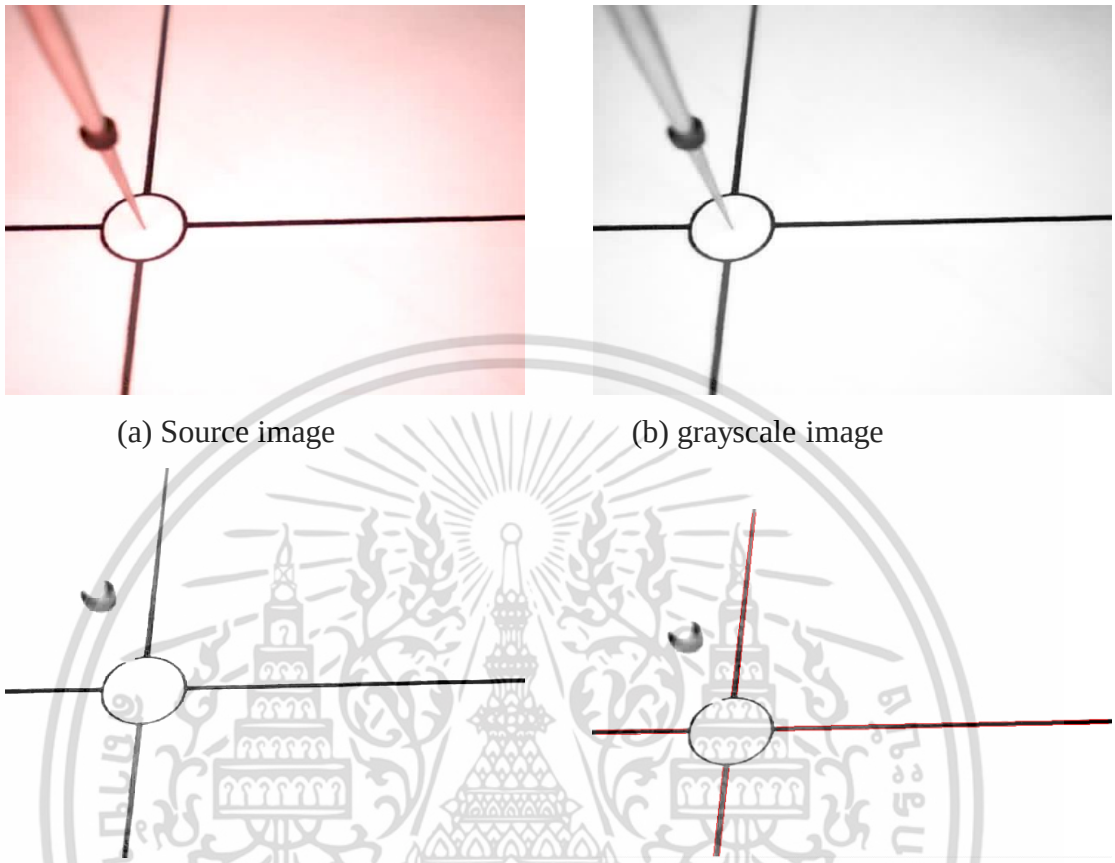


(a) Source image (b) result of IMAQ Detect Lines

Figure C.1 Center point identification out of circular four-arm model

The identification algorithm includes following steps:

1) Acquire image, Fig. C.2a.
2) Convert color image to grayscale image, Fig C.2b.
3) Adjust brightness and contrast of the grayscale image, Fig C.2c.
4) Apply IMAQ Detect Lines, Fig C.2d shows detected lines.
5) Compute middle line of each two parallel lines, intersection point of at least two line that form angle between 72-95 degree is the center point. Fig. C.1b shows the calculated intersection point (i.e. central point of the four-arm model).



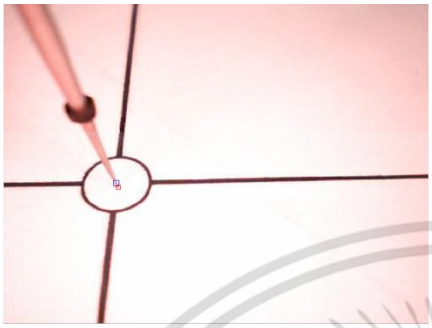
(a) Source image (b) grayscale image

(c) Result after adjust brightness and contrast (d) result of IMAQ Detect Lines

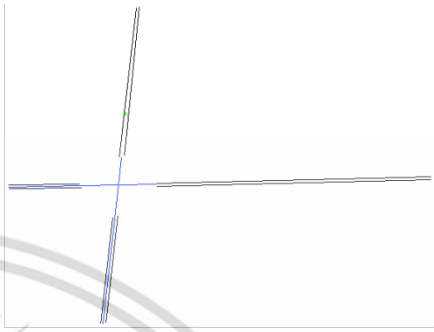
Figure C.2 Processed images for feature extraction out of circular four-arm model

Even though each individual arm is one single thin line, the line extraction produces an appearance of two parallel lines for each arm. The upper and lower middle points that form a line between the two parallel lines of each arm are then computed. The intersection angles of any two arms are between 72 and 95 degrees due to the camera perspective. Other lines with an angle outside the 72 and 95 degrees are removed and regarded as noise or unwanted lines. In the worst-case scenario in which the end-effector obstructs an arm from the camera's view, the intersection point can be calculated from only two lines with an angle between both lines of 72 to 95 degrees. Since at certain angles the probe tip could obstruct some parts of any two arms, the proposed algorithm thus has to be robust in calculating the central control point. With this localization

method, the central control point can be identified at any rotational angle as the robot rotates around the control point.

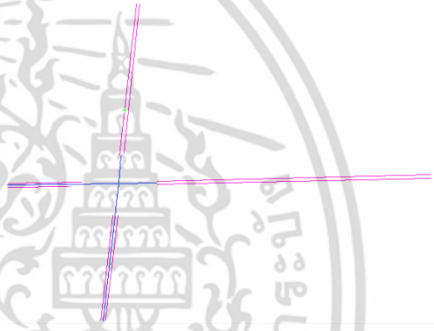


(a) Source image at 0 degree calibration



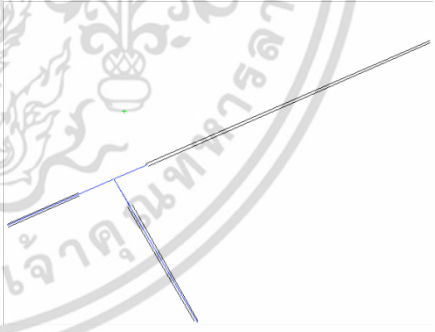
(b) detected lines and center, 0 degree

(c) Source image at 16 degree calibration



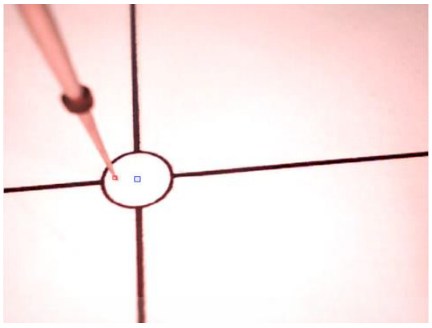
(d) detected lines and center, 16 degree

(e) Source image at 150 degree calibration

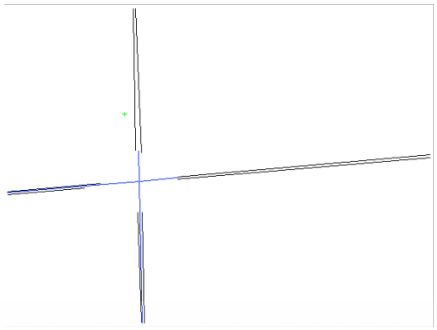


(f) detected lines and center, 150 degree

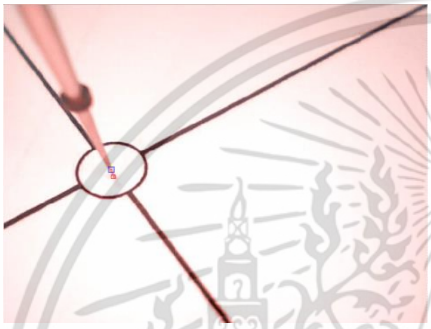
Figure C.3 Sample images and identification results at different degree of calibration



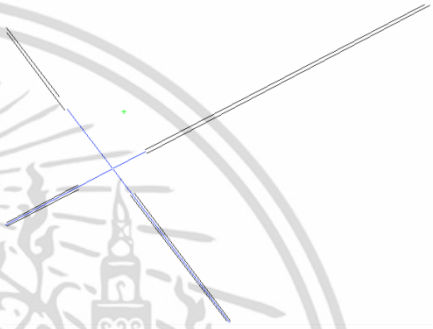
(a) Source image at 174 degree calibration



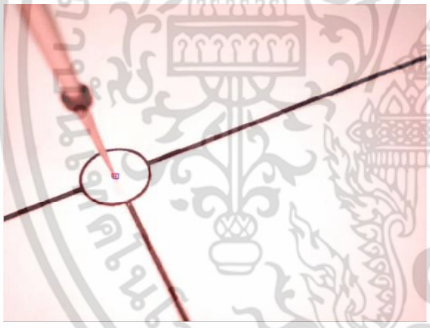
(b) detected lines and center, 174 degree



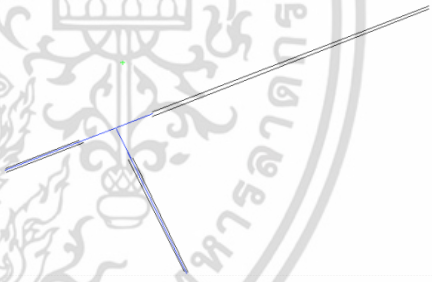
(c) Source image at 235 degree calibration



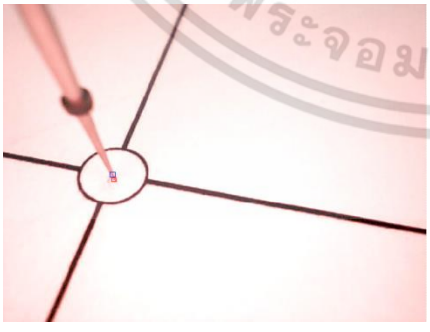
(d) detected lines and center, 235 degree



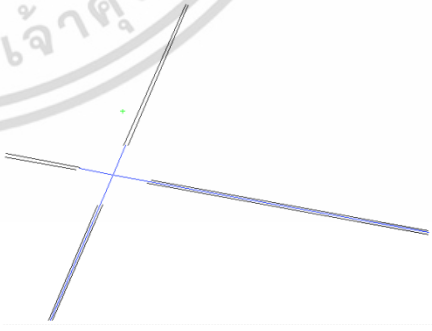
(e) Source image at 244 degree calibration



(f) detected lines and center, 244 degree



(g) Source image at 284 degree calibration



(h) detected lines and center, 284 degree

Figure C.4 Additional sample calibration images and identification

APPENDIX D

Rotational Calibration Data

This sub-section contains the calibration data which is recorded during rotational calibration of the robot's head. The relative position of the X-axis and Y-axis is recorded in unit of micrometers.

Robot θ	X-axis	Y-axis	Robot θ	X-axis	Y-axis
0.00	349409.00	314618.00	31.00	351113.00	306760.00
1.00	349360.00	314321.00	32.00	351226.00	306523.00
2.00	349348.00	314040.00	33.00	351370.00	306299.00
3.00	349362.00	313780.00	34.00	351487.00	306074.00
4.00	349331.00	313501.00	35.00	351635.00	305860.00
5.00	349353.00	313233.00	36.00	351732.00	305640.00
6.00	349376.00	312970.00	37.00	351891.00	305425.00
7.00	349385.00	312717.00	38.00	352051.00	305216.00
8.00	349420.00	312460.00	39.00	352191.00	304998.00
9.00	349435.00	312214.00	40.00	352334.00	304787.00
10.00	349480.00	311951.00	41.00	352509.00	304582.00
11.00	349504.00	311674.00	42.00	352665.00	304373.00
12.00	349557.00	311418.00	43.00	352842.00	304179.00
13.00	349589.00	311167.00	44.00	353020.00	303992.00
14.00	349652.00	310907.00	45.00	353174.00	303799.00
15.00	349694.00	310645.00	46.00	353364.00	303611.00
16.00	349741.00	310396.00	47.00	353536.00	303410.00
17.00	349816.00	310148.00	48.00	353728.00	303233.00
18.00	349869.00	309903.00	49.00	353894.00	303050.00
19.00	349954.00	309650.00	50.00	354075.00	302869.00
20.00	350020.00	309387.00	51.00	354280.00	302700.00
21.00	350112.00	309144.00	52.00	354454.00	302527.00
22.00	350182.00	308892.00	53.00	354635.00	302361.00
23.00	350286.00	308643.00	54.00	354840.00	302208.00
24.00	350341.00	308391.00	55.00	355035.00	302039.00
25.00	350450.00	308152.00	56.00	355253.00	301890.00
26.00	350563.00	307919.00	57.00	355450.00	301729.00
27.00	350625.00	307693.00	58.00	355643.00	301580.00
28.00	350748.00	307457.00	59.00	355850.00	301429.00
29.00	350874.00	307225.00	60.00	356055.00	301285.00
30.00	350983.00	306982.00	61.00	356259.00	301153.00

Robot θ_i	X-axis	Y-axis
62.00	356487.00	301029.00
63.00	356683.00	300904.00
64.00	356899.00	300773.00
65.00	357137.00	300658.00
66.00	357351.00	300538.00
67.00	357560.00	300423.00
68.00	357781.00	300313.00
69.00	358027.00	300215.00
70.00	358212.00	300106.00
71.00	358456.00	300018.00
72.00	358669.00	299924.00
73.00	358925.00	299843.00
74.00	359154.00	299751.00
75.00	359385.00	299666.00
76.00	359634.00	299600.00
77.00	359865.00	299521.00
78.00	360100.00	299446.00
79.00	360356.00	299392.00
80.00	360587.00	299324.00
81.00	360813.00	299264.00
82.00	361077.00	299223.00
83.00	361333.00	299164.00
84.00	361578.00	299113.00
85.00	361821.00	299063.00
86.00	362089.00	299040.00
87.00	362341.00	299000.00
88.00	362587.00	298970.00
89.00	362845.00	298961.00
91.00	363359.00	298905.00
92.00	363622.00	298911.00
93.00	363890.00	298899.00
94.00	364147.00	298914.00
95.00	364408.00	298910.00
96.00	364672.00	298888.00
97.00	364932.00	298917.00
98.00	365183.00	298925.00
99.00	365427.00	298941.00
100.00	365690.00	298985.00
140.00	375198.00	303752.00
141.00	375367.00	303955.00
142.00	375532.00	304135.00
143.00	375690.00	304340.00
144.00	375856.00	304523.00
145.00	376011.00	304740.00
146.00	376182.00	304945.00
147.00	376326.00	305162.00
148.00	376483.00	305358.00
149.00	376635.00	305562.00
150.00	376770.00	305790.00
151.00	376913.00	305995.00
152.00	377037.00	306222.00
153.00	377172.00	306426.00
154.00	377292.00	306664.00
155.00	377430.00	306897.00
156.00	377538.00	307133.00
157.00	377661.00	307356.00
158.00	377781.00	307585.00
159.00	377880.00	307831.00
160.00	377991.00	308054.00
161.00	378079.00	308296.00
162.00	378182.00	308525.00
163.00	378262.00	308779.00
164.00	378365.00	309031.00
165.00	378433.00	309281.00
166.00	378522.00	309528.00
167.00	378604.00	309777.00
168.00	378662.00	310035.00
169.00	378734.00	310275.00
170.00	378779.00	310529.00
171.00	378853.00	310769.00
172.00	378890.00	311032.00
173.00	378949.00	311303.00
174.00	378999.00	311561.00
175.00	379023.00	311816.00
176.00	379072.00	312074.00
177.00	379089.00	312338.00
178.00	379125.00	312587.00
179.00	379159.00	312843.00
180.00	379183.00	313097.00
181.00	379205.00	313363.00
182.00	379197.00	313626.00
183.00	379210.00	313894.00
184.00	379215.00	314158.00
185.00	379193.00	314425.00
186.00	379191.00	314681.00
187.00	379161.00	314941.00
188.00	379153.00	315194.00
189.00	379114.00	315446.00

Robot θ_i	X-axis	Y-axis
190.00	379097.00	315709.00
191.00	379066.00	315981.00
192.00	379012.00	316236.00
193.00	378981.00	316489.00
194.00	378917.00	316749.00
195.00	378875.00	317003.00
196.00	378827.00	317246.00
197.00	378752.00	317494.00
198.00	378704.00	317737.00
199.00	378641.00	317991.00
200.00	378573.00	318250.00
201.00	378480.00	318493.00
202.00	378408.00	318741.00
203.00	378328.00	318988.00
204.00	378221.00	319231.00
205.00	378140.00	319468.00
206.00	378029.00	319701.00
207.00	377942.00	319929.00
208.00	377839.00	320176.00
209.00	377726.00	320425.00
210.00	377598.00	320651.00
211.00	377467.00	320872.00
212.00	377352.00	321109.00
213.00	377230.00	321338.00
214.00	377085.00	321556.00
215.00	376960.00	321779.00
216.00	376835.00	321994.00
217.00	376676.00	322208.00
218.00	376533.00	322436.00
219.00	376371.00	322639.00
220.00	376224.00	322854.00
221.00	376050.00	323058.00
222.00	375900.00	323267.00
223.00	375745.00	323467.00
224.00	375566.00	323654.00
225.00	375387.00	323836.00
226.00	375216.00	324040.00
227.00	375034.00	324243.00
228.00	374841.00	324419.00
229.00	374663.00	324607.00
230.00	374482.00	324791.00
231.00	374277.00	324960.00
232.00	374091.00	325136.00
233.00	373885.00	325293.00
234.00	373700.00	325460.00
235.00	373483.00	325615.00
236.00	373279.00	325788.00
237.00	373074.00	325953.00
238.00	372858.00	326091.00
239.00	372629.00	326231.00
240.00	372417.00	326387.00
241.00	372207.00	326532.00
242.00	371980.00	326656.00
243.00	371771.00	326792.00
244.00	371533.00	326911.00
245.00	371296.00	327053.00
246.00	371060.00	327184.00
247.00	370825.00	327287.00
248.00	370591.00	327412.00
249.00	370344.00	327510.00
250.00	370112.00	327624.00
251.00	369873.00	327732.00
252.00	369631.00	327814.00
253.00	369379.00	327919.00
254.00	369127.00	328017.00
255.00	368877.00	328087.00
256.00	368629.00	328176.00
257.00	368369.00	328239.00
258.00	368110.00	328322.00
259.00	367866.00	328394.00
260.00	367612.00	328442.00
261.00	367367.00	328505.00
262.00	367103.00	328566.00
263.00	366842.00	328601.00
264.00	366582.00	328654.00
265.00	366327.00	328679.00
266.00	366060.00	328701.00
267.00	365807.00	328744.00
268.00	365553.00	328779.00
269.00	365294.00	328786.00
270.00	365042.00	328811.00
271.00	364778.00	328831.00
272.00	364503.00	328844.00
273.00	364244.00	328832.00
274.00	363979.00	328835.00
275.00	363712.00	328813.00
276.00	363456.00	328811.00
277.00	363203.00	328801.00

Robot θ_i	X-axis	Y-axis
278.00	362947.00	328765.00
279.00	362698.00	328749.00
280.00	362435.00	328703.00
281.00	362163.00	328676.00
282.00	361898.00	328644.00
283.00	361648.00	328587.00
284.00	361395.00	328545.00
285.00	361139.00	328497.00
286.00	360886.00	328425.00
287.00	360637.00	328373.00
288.00	360394.00	328296.00
289.00	360140.00	328253.00
290.00	359891.00	328164.00
291.00	359647.00	328072.00
292.00	359408.00	327976.00
293.00	359160.00	327898.00
294.00	358917.00	327813.00
295.00	358679.00	327745.00
296.00	358446.00	327632.00
297.00	358218.00	327517.00
298.00	357982.00	327393.00
299.00	357732.00	327280.00
300.00	357506.00	327151.00
301.00	357275.00	327041.00
302.00	357046.00	326901.00
303.00	356829.00	326782.00
304.00	356620.00	326666.00
305.00	356401.00	326540.00
306.00	356193.00	326391.00
307.00	355977.00	326254.00
308.00	355768.00	326093.00
309.00	355544.00	325939.00
310.00	355345.00	325775.00
311.00	355127.00	325617.00
312.00	354929.00	325440.00
313.00	354720.00	325284.00
314.00	354532.00	325106.00
315.00	354337.00	324948.00
316.00	354149.00	324757.00
317.00	353948.00	324573.00
318.00	353772.00	324381.00
319.00	353583.00	324201.00
320.00	353397.00	324016.00
321.00	353228.00	323813.00
322.00	353066.00	323608.00
323.00	352891.00	323419.00
324.00	352727.00	323231.00
325.00	352572.00	323014.00
326.00	352401.00	322804.00
327.00	352241.00	322596.00
328.00	352103.00	322379.00
329.00	351946.00	322168.00
330.00	351810.00	321939.00
331.00	351663.00	321729.00
332.00	351522.00	321516.00
333.00	351404.00	321290.00
334.00	351284.00	321052.00
335.00	351143.00	320819.00
336.00	351013.00	320590.00
337.00	350909.00	320354.00
338.00	350789.00	320119.00
339.00	350691.00	319872.00
340.00	350581.00	319646.00
341.00	350494.00	319402.00
342.00	350370.00	319193.00
343.00	350289.00	318939.00
344.00	350215.00	318688.00
345.00	350123.00	318434.00
346.00	350060.00	318185.00
347.00	349976.00	317935.00
348.00	349899.00	317681.00
349.00	349828.00	317433.00
350.00	349784.00	317178.00
351.00	349722.00	316938.00
352.00	349661.00	316678.00
353.00	349603.00	316408.00
354.00	349575.00	316150.00
355.00	349529.00	315899.00
356.00	349508.00	315632.00
357.00	349463.00	315367.00
358.00	349432.00	315115.00
359.00	349425.00	314856.00

PUBLICATIONS

[1] T. Banlue, P. Sooraksa and S. Noppanakeepong, "A Practical Visual Servo Calibration of End-Effector Pose Estimation in Industrial Manufacturing," *International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, pp. 112-116, Nov. 2012.

[2] T. Banlue, P. Sooraksa and S. Noppanakeepong, "Non-linear Visual Servo System for Printed-Circuit Board Fault Injection Test," *The Fifth Asia-Pacific Workshop on Chaos Control and Synchronization (APWCCS)*, Kunming, China, Oct. 2010.

[3] T. Banlue, P. Sooraksa and S. Noppanakeepong, "A Practical Position-Based Visual Servo Design and Implementation for Automated Fault Insertion Test," *International Journal of Control, Automation, and Systems (IJCAS)*, vol. 12, no. 5, Oct. 2014.

VITA

Tassanai Banlue was born in Ubon Ratchathani Province to a family of farmers. He received B.Ind and M.Eng, both in Electrical Engineering. Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand, in 1998 and 2002, respectively. He has been working with Celestica (Thailand) Ltd as a test engineer since 2002. He has designed and implemented automatic test solutions for the testing of electronic products for a number of customers such as Oracle, IBM, Alcatel-Lucent, EMC and Cisco.

He has been a pioneer developer of vision systems at Celestica since 2009. A year later, he developed visual inspection projects, an automatic insertion machine and Cartesian robot. His experience and knowledge of vision systems and robotics are merged for the proposal of the visual servo system.

His main field interests include Automatic testing, Image Processing, Machine Vision, Wireless Communications, RF and Magnetic Resonance Wireless Power Transfer. Recently, he researched and prototyped a wireless power and energy harvesting product. He is currently living in Chonburi province. He combined full time employment with his desire to study D.Eng at KMITL. He encountered great difficulty supporting his wife and small child whilst also working fulltime at the company as a project lead and technical consultant.