ANALYSIS OF SHAPE RECOGNITION USING ROBOTIC FINGERS BY IMAGE PROCESSING TECHNIQUE FOR OBJECT MANIPULATION

MUHAMAD AQMAL BIN MUSA

BACHELOR IN MECHATRONICS ENGINEERING
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

ANALYSIS OF SHAPE RECOGNITION USING ROBOTIC FINGERS BY IMAGE PROCESSING TECHNIQUE FOR OBJECT MANIPULATION

MUHAMAD AQMAL BIN MUSA

A report submitted
in partial fulfillment of the requirements for the degree of
Mechatronics Engineering

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled “Analysis of Shape Recognition Using Robotic Fingers by Image Processing Technique For Object Manipulation” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : ....

Date : ....

APPROVAL

I hereby declare that I have checked this report entitled "Analysis of Shape Recognition Using Robotic Fingers by Image Processing Technique For Object Manipulation" and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honors

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to Almighty Allah for enabling me to complete the study at Universiti Teknikal Malaysia Melaka (UTeM). It is always in pleasure to remind the fine people in the Faculty of Electrical Engineering for their sincere guidance I received to uphold the study as well as theoretical and practical skills in engineering.

Firstly, I would like to thank Prof. Datuk Ts. Dr. Shahrin bin Sahib (Vice Chancellor, UTeM) for meticulously planning academic curriculum in such a way that students are not only academically sound but also industry ready by including such industrial training and final year project patterns. I would also like to acknowledge and heartfelt gratitude to Prof. Madya Dr. Ahmad Zaki bin Hj. Shukor (FYP Supervisor) and the other lecturers involved (FYP Committee) as well as the panels for keep and never give up on guide the students for allowing the final year project going smoothly.

Last but not least, a special thanks to my parents, Mr. Musa bin Khamis, Mrs. Nor Asmah binti Abdullah and my family members for always support me in terms of motivational, transportation, financial, and others.

ABSTRACT

In line with the advancement of technologies, present tactile sensing based gripping technologies have shown improvements in interactive force gripping and identification of object. However, the precision of shape recognition using tactile sensors on gripper is unsatisfied. The main objective for this study is to design and develop a system for identifying the contact points of different object shapes and analyze the angular motion of fingers movement as well as accuracy of object grasped. The design of gripper involves two robotic fingers with two degree of freedom. It implements basic image processing techniques such as gray-scale, threshold and edge detection as the approach to determine the vertices on the object image for recognizing the object's shape. Plus, the contact points on the object surface that will be touched by the gripper's fingers define the orientations of the joints by using inverse kinematic equation. The effectiveness of this vision system required large size of targeted object compared to smaller size. These analyses obtained 2.05 and 2.0 pixels per millimeter which have been converted from pixel coordination to the real coordination for circle object and square object respectively. The accuracy on the orientations calculated in actual value and pixel value were just in slightly different which were 97.54% (Link 1) and 97.4% (Link 2) for grasping circle object while 97.48% (Link 1) and 93.8% (Link 2) for grasping the square object.

ABSTRAK

Sebaris dengan kemajuan teknologi, sensor sentuhan kini yang berasaskan teknologi pencengkam telah menunjuk penambahbaikan bagi daya interaktif mencengkam dan mengenalpasti objek. Objektif utama kajian ini adalah untuk mereka bentuk dan membangunkan suatu sistem untuk mengenal pasti titik hubungan pada bentuk objek yang berbeza dan menganalisis pergerakan jari serta ketepatan objek yang dicengkam. Reka bentuk penggenggam melibatkan dua jari robot dengan dua darjah kebebasan. Kajian ini melaksanakan teknik pemprosesan imej asas seperti skala kelabu, ambang dan pengesanan pinggir sebagai pendekatan untuk menentukan jumlah bucu untuk mengenali bentuk objek. Tambahan pula, titik hubungan pada permukaan objek yang akan disentuh oleh jari-jari penggenggam menentukan orientasi pada sendi dengan menggunakan persamaan kinematic songsang. Keberkesanan sistem penglihatan ini memerlukan saiz yang besar bagi objek yang disasarkan berbanding saiz yang lebih kecil. Analisis ini memperoleh 2.05 dan 2.0 piksel bagi setiap millimeter yang telah ditukarkan daripada koordinasi piksel ke koordinasi sebenar. Ketepatan pada orientasi yang dikira dalam nilai sebenar dan nilai piksel hanya sedikit berbeza iaitu 97.54% (Link 1) dan 97.4% (Link 2) untuk menggenggam objek bulat manakala 97.48% (Link 1) dan 93.8% (Link 2) untuk menggenggam objek segi empat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter briefly describes about motivation, problem statement, project objectives and limitations of the project. Motivation inspired to current manufacturing industry development. Problem statement in this report written referring to the information gathering after studied the previous researches. As well as objectives and scopes for this project, it was set by identifying the problems.

1.2 Motivation

In this era of globalization, industrial sector plays a main role in developing the country. There are several broad categories of industry, namely genetic industries, extractive industries, manufacturing industries and construction industries. In addition, genetic and extractive industries are generally known as primary industries due to their output of raw materials such as forestry, farming, fishing and mining. Meanwhile, the manufacturing and construction industries are known as secondary industries because of their output of finished products that are converted from raw materials such as goods, electrical equipment, vehicle and building.

In fact, the developing countries and underdeveloped countries are making terrific developments for growth of the manufacturing industries. In the large scale of the manufacturing industry, automation and robotic are implemented. It shown in the Figure 1.1, the statistic represents worldwide sales of industrial robots between 2004 and 2017 announced at Automatica 2018 in Munich by International Federation of Robotic (IFR). It shows that the industrial robots sales increased by 29 percent from about 294,300 units sold in 2016 to around 387,550 units sold in 2017 [1]. It can be observed, that every year for the previous decade is showing increasing sales of industrial robots.

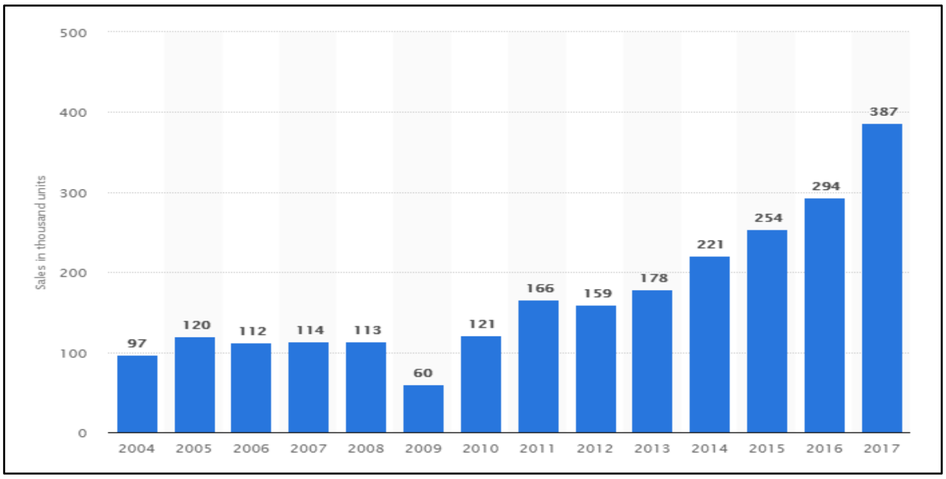


Figure 1.1: Worldwide Sales of Industrial Robots 2004-2017 (in 1,000 Units) [1]

In the robotics industry, mechanical arm is a popular application. However, arm is useless without hand. The robot end-effector is the connection between the robot arm and the environment around it. Generally, the implementation of robotic end-effector is for part handling. In industrial, there are many application of it. Collaboration of other field in engineering such as mission vision, artificial intelligence, control system and others can be made to improve the robotic end-effector. As shown in Figure 1.2, the graph represents the total machine vision financial transactions by North America from 2001 to 2015 stated in Shanghai by Automated Imaging Association (AIA). It shows that the market growth by 13 percent on average since 2010 [2]. It can be determine, machine vision and industrial robot are the technologies that corresponding in speedy developing until now.

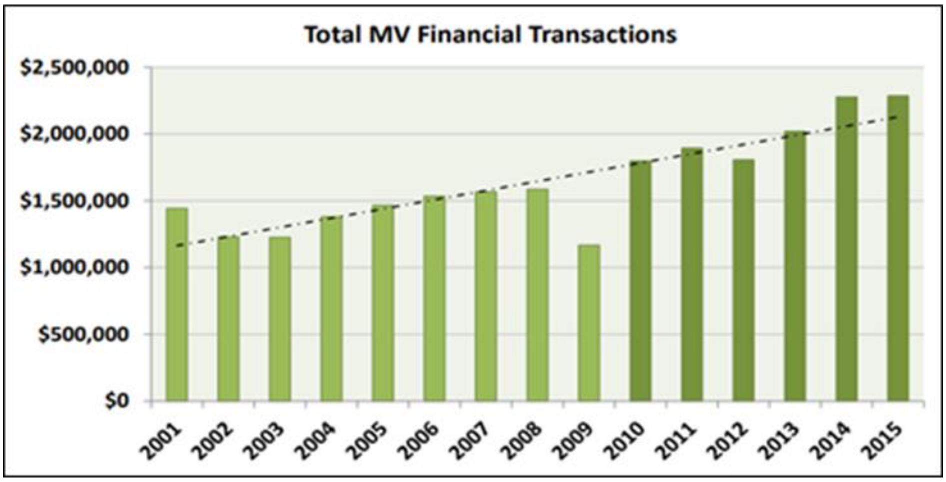


Figure 1.2: North America Total Machine Vision Financial Transaction from 2001 to 2015 (in USD) [2]

In conclusion, the substitution of vision sensor as human eyes to provide visual system for the machine is popular nowadays. It is applied for inspection, measurement, recognition and also determines location. So that, it is important to provide a study related to the collaboration of machine vision and robotic gripper. It recognizes the targeted objects first and then does the gripping mechanism. This kind of study may help to initialize the steps before grasping, prevent the insecurities of the object and give massive other benefit in future.

1.3 Problem Statement

This section elaborates on common problems faced by the user of robotic hand or gripper. There is numerous problems state in the use of gripper. Then, there are various ways to improve the use of gripper in the industry. First of all, the gripper has no common design suitable for all uses. It has a wide range of designs applied in the industry. Each of the design has its own use. It is for specific tasks and do not has diversity in its application. Most of the wide-ranging design in the industry has a simple and modifiable design.

However, there are few studies that involve the use of gripper other than the purpose of grasping. In instance, the gripper can be used to define the position and the shape of grasped object. The collaboration along with other fields such as machine learning, machine vision, sensors and etc. can be made to create a multifunctional gripper. As we know that the complex design and functional gripper is the end-effector that attached to robot manipulator or robotic arm. Most of previous researches were made for the purpose of studied for identifying the positions of the object involving robotic arm. There is relationship has been discovered to determine the effectiveness of usage of robot manipulator which is the amount of degree-of-freedom attached to the robot. It costs a lot, difficult to get and using complex processes.

Finally, the basic gripper is an open loop system and simple. Unless, it involves the measurement acts as feedback such a haptic feedback then it will be a closed loop system. Gripper that applied haptic feedback touch the object first and then it measure variables needed such as force, shape and position parameter. In this case, the haptic feedback is substitute by the vision system. Undeniably, layout of the processes in the field of computer vision is difficult to match with the gripping mechanism. Moreover, this kind of system applied is inversely to the haptic feedback. It measures the contact points needed before it touches the targeted object.

1.4 Objective

The main purposes of this project are to:

1. Design and develop a planar gripper using two robotic fingers arrangement.

2. Develop machine vision system for identifying the contact points of the object that will be grasped by the robotic fingers.

3. Analyze angular motion of gripper to grip object and determine accuracy of object grip using repeatability test.

1.5 Scope

The limitations of this project are:

1. Shape recognition by using two moveable fingers of robotic gripper applied with a webcam as imaging device.

2. It applied two degree-of-freedom of robotic fingers and location of a webcam is on top of the workspace.

3. 420 mm x 594 mm (A3 size) of grid paper is the area cover of the workspace.

4. Light intensity never set because it used the default light in lab environment and the position of imaging device is fixed.

5. Consider only two shapes of the objects which are circular and square.

6. Approximately $50.27cm^2$, $28.27cm^2$ and $12.57cm^2$ size of circular and 64 cm^2 size of square for target objects.

7. Implementation of OpenCV-Python as an image processing platform and Arduino as a controller for gripping mechanism.

8. Two moveable fingers of gripper will be in initial position before the program start.

9. Two moveable fingers of gripper move in determined angular after the program recognized the shape of the object.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Robotic end-effector and machine vision are the technology that is empowered to this day. Both are widely used in manufacturing industry. The collaboration between these two technologies involved in inventing a gripper that is able to identify the shape of an object. The project approach is to substitute the human gripping and visioning functions. Further information is divided into two parts, the techniques used to identify the visual features and exactly how to grip the targeted object.

2.2 Mechanical Gripper

Generally, implementation of robotic end-effector is for part handling. There are numerous of end-effector in designs such as hook, vacuum, mechanical gripper, clamp, scoop, magnetic and etc. Most popular end-effector used in manufacturing industry is mechanical gripper. In addition, there are several things that need to be emphasized on the use of it which are design and external factors. The designed involves the amount of finger usage, degree of freedom, mechanism, force applied and drive method; electric, pneumatic or hydraulic. Moreover, external factors such as slippage, friction, compliance and dynamic also affect the use of it.

2.2.1 Design

There are a few things that need to be considered in the implementation of robotic hand. First of all, most priority refers to the drive method applied for it. In engineering there are three drive methods which are electric, pneumatic and hydraulic. All of previous studies [1]–[15] used electrical drive except [16] which uses pneumatic drive. In instance, research [10] used dc motor and research [11], [17] used servomotors as their drive.

Secondly, the thing that needs to be considered for the selection of gripper is the amount of fingers use. Frankly, at least two fingers used for gripping purpose. Most previous researches implemented two fingers in their cases [3], [7]–[9], [11], [17]. There are many designs for two fingers gripper such as four-bar link gripper [11], jaw gripper [3], [7], parallel gripper [8], and clamp gripper [7], [17].

Next, researchers for articles [5], [12], [14] have three fingers implemented for their gripper. It is also known as Barrett hand. In articles [12], [14], the researchers stated that they used two movable fingers oppose to the fixed thumb and all fingers are attached to the palm. The grasping system is even more complex if it applies more than two fingers. It illustrates in Figure 2.1, the Barrett hand is used in the previous research [14]. Then, the studies used four fingers called dexterous hand for the gripper design [6], [16]. It mostly designs same like human hand but with four fingers. Meanwhile, a few studies are using an anthropomorphic design which includes five fingers and a palm [10], [13]. It is a complex design that is exactly same like human hand.

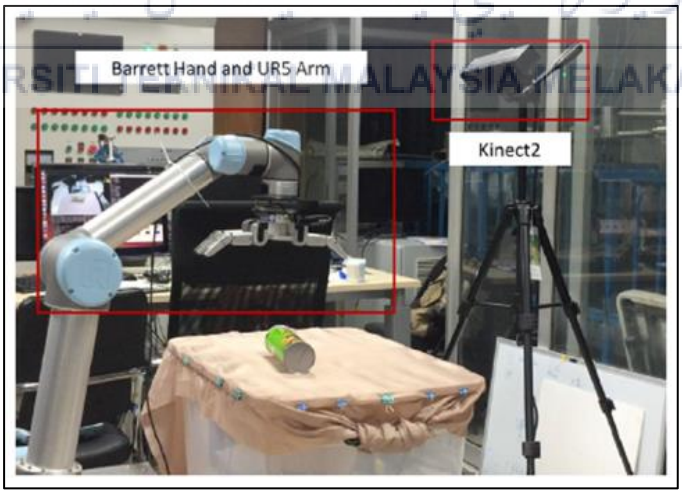


Figure 2.1: The Experimental Setup Involved Barrett Hand [14]

2.2.2 Degree of Freedom

Another important thing that needs to be considered for gripper selection is the degree of freedom (DOF) implemented by each finger used. Most of the jaw, parallel or clamp gripper designs are 1 DOF [3], [7], [11], [14], [17]. It used only a basic mechanism such as close and open to perform grasped. It only controls the rotation of the motor by rotate it clockwise and anticlockwise rotation. Meanwhile, there are grippers that implemented the robotic fingers with more than 1 DOF. Most of the previous studies used 2 DOF for their robotic fingers [5], [8]–[10], [16]. Other than that, the dexterous hand and anthropomorphic are use 3 or 4 DOF [6], [12], [13]. In fact, the hypothesis stated the more usage of degree of freedom for the robots then the more accurate and precise the task performed. Thus, the mechanism and mathematical derivation applied will be more complicated.

The degree of freedom applied in each link needs to be considered about the mechanism used. Basically, the kinematic equation derived for the robotic fingers that fulfilled by more than 1 DOF. The kinematic is the study of motion that involves position, velocity or acceleration. The previous studies applied the forward kinematic in their research for mapping from Joint space to Cartesian space [3], [6], [9], [12]. In easiest way for understanding, it is the derivation of finding the position of the end of link by given the orientation of the joint of links. Whereas, the previous studies were involved the inverse kinematic in their research are mapping the Cartesian space to Joint space [13], [17]. It stated that the progression of finding the desire orientations by using the reference position. Different with the article [10], it is studied about the kinematic redundancy. It is the study about the needed of additional degree of freedom by adding the extra joints and links to the mechanism.

2.3 Machine Vision

Machine vision is also called as robotic vision or computer vision. It has countless techniques and implementations made for inspection, identification and navigation purposes. Most of the applications in industrial are inspection while in research and learning is more focused on identification. As shown in Figure 2.2, the techniques involved in machine vision systems can be classified in three, namely image acquisition, image processing and image analysis [18]. Each class has its specific important role to be explained in the next subdivision.

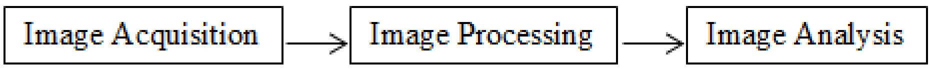


Figure 2.2: Machine Vision Block Diagram [18]

2.3.1 Image Acquisition

The first stage in machine vision is called image acquisition or image formation. This process known as low level process in machine vision system that determined input and output of this process are in image form. Image acquisition is the sequential process of image sensing, image digitizing and image preprocessing. Figure 2.2 shows the basic elements in the low level process [19]. The visual information that can be determined by image is color, geometry (shape, position, dimension, density and texture) and movement (dynamic process). Theoretically, determinant of contrast and sharpness are the keys in image sensing. So that, illumination technique used and imaging device selection play an important role. Most illumination techniques used to provide good contrast in previous researches are front light and clear background to differentiate and focus on targeted object. Front light is an external light source provided by other device such as LED spot light, LED ring light or fluorescence lamp to help in improves the enlightenment.

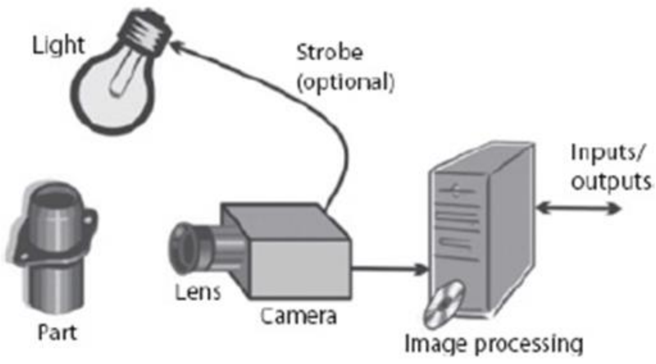


Figure 2.3: The Basic Elements in Image Acquisition [19]

Imaging device is also known as camera. There are several things that need to consider for camera selection which are resolution, field of view, lens, light, signal-to-noise ratio, stability and cost. It has various types in market with different specifications for achieved good sharpness of the image. This device helps in conversion of real image to synthetic image (digital image) whether for photo snapping or video recording. Basic imaging device used in previous researches is monocular stereoscope [4]–[7], [15]. It is a one-eye visualization that can only provide two-dimensional image. A RGB-camera is the most popular basic camera used for research. In addition, other type of camera used in previous researches called stereo camera [8]. Stereo camera is the type of camera with two or more lenses with a distinct image sensor or film frame for every frame. It also known as binocular stereoscope [7]; it is two-eye visualization that has ability to provide three-dimensional image (stereo photography) and basically it provide information occupied depth. Several models used in previous researches which are RGB-D camera [3], Kinect RGBD camera [12], [14], Bumblebee2 camera [10], [11] and two pairs of Sony B/W cameras [9].

Figure 2.4: The Example of Stereo Camera [12]

In digitizing image, two main elements involve called signal conditioning and digital storage. No information provided by the previous researches because this processes are ready built-in in the imaging device or controller. Mostly, it involves the visual information that converted to electrical signal by camera. The electrical signal is in analog formed converts to digital signal by analog-to-digital signal converter. The converted signal will be stored in frame grabber which is the image storage. After that, digitized image will be passing to preliminary process (pre-processing) for further image processing technique.

The last process in low level process of machine vision is image preprocessing. This process aimed the techniques used to improve digital image by eliminate unwanted distortions or enhances some essential image features for further process. Basically, it also can be determined as enhancement; it is used for noise removing, de-blurring, smoothing, sharpening or brightening the input image. Previous research in [17] used noise reduction technique called anti-aliasing. It is a technique of minimizing the distortion artifacts known as aliasing when representing a high resolution image at a lower resolution. In easier way for understanding, it is a process of removing signal components that have higher frequency.

Other than that, research made in [11] used retina-inspired structure called as novel filter. This technique makes the improvement in sharpness for the visual stimulus. There are also the studies using a basic technique called as histogram equalization [4], [7], [15]. It converted the image data to gray-scale value. This gray-level system provides the data in gray tone which has 256 different values (0 until 255) that can be manipulate. In addition, three-dimensional image construction involved in [3], [10], [16] used visual shape primitive to make the collection of data measured by pointing on the surface of the object sensed called as point cloud. Research in [16] also used image restoration inverse problem which is 3D-filtering for noise reduction and de-blurring. Stated in the previous researches [4], [15], this process helps in accuracy improvement for the image processing techniques.

2.3.2 Image Processing

In machine vision system, image processing is the mid-level process that has the numerous of techniques already been explored. This process received an input which is in image form and produced attributes as an output. It can be categorized into two categories which are segmentation and object recognition. Firstly, main role of the image segmentation is partitioning a digital image into multiple segments. It is also called the features extraction. It can be divided into two types; threshold and edge detection.

Threshold is the conversion process of gray-scale image (black and white) to binary image (0 and 1). In articles [6], [11]–[14], [17], they used histogram equalization to perform global threshold for providing important features such as color, intensity or texture. It also applied in differentiating the targeted object from the background. The authors for article [16] describe in details about the color-based segmentation collaborates with MATLAB software involved in their research. Moreover, there is the technique that implemented the threshold which is Otsu's method. This technique that was explained in articles [9], [12] is the technique for reducing the large gray-level image to small binary image. It uses to automatically execute clustery-based image threshold. Figure 2.5 shows the example of application in using threshold technique.

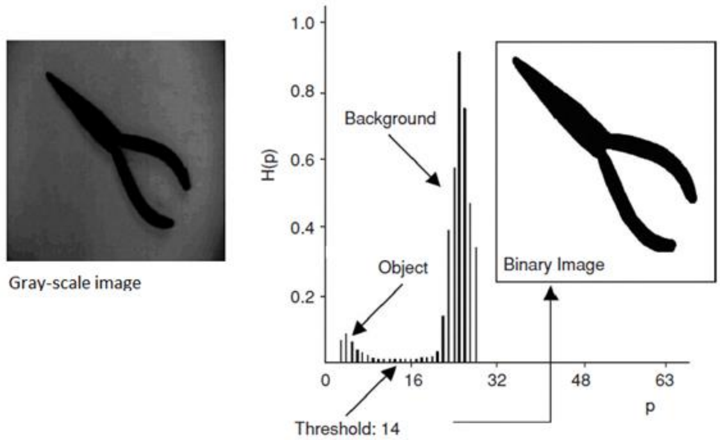


Figure 2.5: The Example Implementation Threshold Technique [20]

However, other technique uses in segmentation process is edge detection. It determines the intensity change at the boundary in an image. Method used in the articles [4], [7] is canny detector. It is the operation of edge detection that uses a multistage algorithm to recognize the wide range of edges in image. Different with the technique used in articles [3], [15], it used boundary box technique when it involved the depth which is determines in 3-D image. The technique uses to find minimum area boundary box to construct convex closure for the edge detection. The researcher for [13] introduced the technique named Sobel-Feldman operator. It also known as Sobel filter, it is principally within edge detection algorithms where it constructs an image underlining edge.

There are useful processes in image processing named morphological process and object recognition. This process can replace the segmentation process. It process an image based on form, structure or shape. Articles [4], [17] used the technique called opening and closing morphological. Two basic things in morphological process which are dilation and erosion combined to perform a complex operation. It stated in that articles, opening morphological used to remove the useless small object and closing morphological used to remove small holes by fill in the gap. Besides, there is a technique related to topology called object recognition that can be classified with morphological process. In previous researches [3], [6], [9], [16], they used geometric feature extraction to recognize the object. This geometric mode construction is a dimensionality reduction process collaborated with machine learning process to collect geometric features from image. Figure 2.5 shows the implementation of this technique [6].

Figure 2.6: The Example of Geometry Feature Extraction [6]

2.3.3 Image Analysis

Regarding the last process involved in machine vision, the high level of machine vision used to convert the attribute provided towards understanding output called image analysis. There are massive in number of image analysis used in researches. The researches [3], [16] use mapping relationship for their image analysis method. It uses mathematical model called random sample consensus. It implements random sampling to subset data by using MATLAB software for determining the shape identification. Other than that, researches in [14], [15] use Intersection over Union (IoU) threshold as projected cluster model object detection for defining the classifier to be true or false.

Furthermore, there are researches used the labeling technique [14], [17]. It is the algorithm created in MATLAB software for set up the method of classifying the attached components in an image and allocating each one a unique label. Then, article [9] uses Linear Quadratic Estimation also known as Kalman filter. This method is useful in image analysis; a repetitive mathematical process to speedily evaluate the true value, position, velocity or other variety of data. The researchers also use Binary Space Partitioning tree structure (BSP tree). This method is rendering for spatial information by recursively splitting a space into convex sets by hyper planes.

However, majority of previous studies [4], [7], [8], [10], [12], [14] used the image analysis that related to artificial intelligence. Theoretically, machine learning is the course discovered under the artificial intelligence as well as deep learning. But, deep learning is subset of machine learning. Figure 2.7 illustrates the theory stated for easier understanding. It can be interpreted that deep learning falls under machine learning while both of it fall under artificial intelligence.

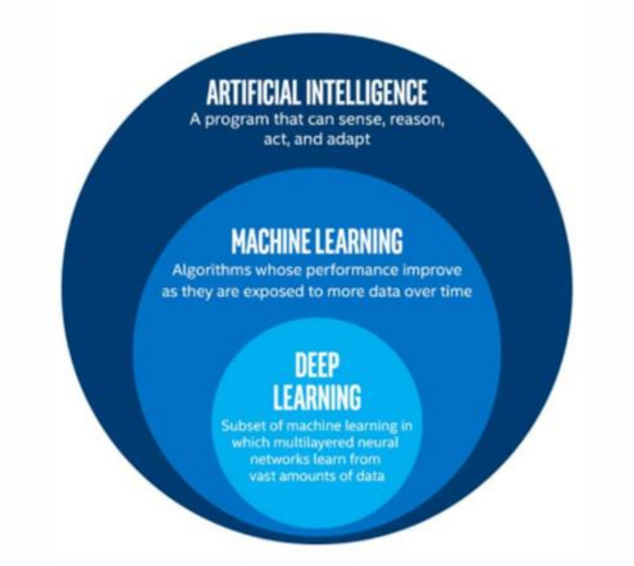


Figure 2.7: The Artificial Intelligence Field

Articles [7], [10] apply Support Vector Machine (SVM) as their image analysis approached. SVM can be classified in machine learning approach because it applied mathematical models or algorithms to classify and perform analysis. The researchers for study [7] state that they applied fuzzy clustering technique. It also falls under machine learning. It identifies the similarity measure to form more than one cluster in which each of data point can be allocated.

Image analysis collaborated with deep learning is applied in [4], [7], [8], [12], [14]. Deep learning is different form machine learning because it is not an algorithm and no mathematical derivation involved. Existence of neural network in deep learning is helpful for carrying studies to succeed. It inspires the nervous system working on human and animal brains for modeling a computer system. It is a process of passing the result to next layer by next layer to the classifier. This feed-forward network applied for union of machine learning algorithms and data inputs complex processing. Figure 2.7 shows the implementation of deep learning in the previous study [14].

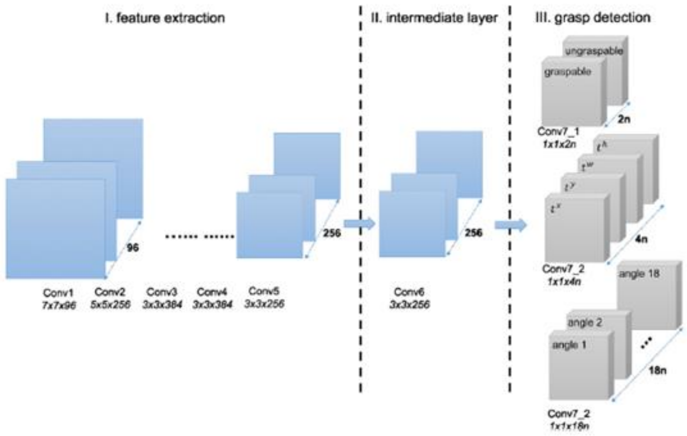


Figure 2.8: Deep Learning Applied in Study [14]

Besides that, some researches fully rely to additional outside tools to perform image analysis. Research [8] stated that IBM Watson is useful. By the way, this tool implemented deep learning to analyze images for scenes, objects, faces and other contexts. Same as research [5], the project used servo visualizing method. It needs an outside support tool to provide feedback information extracted from an imaging device to perform next task. Article [13] used Gazebo simulator also the additional outside tool for providing the shape information after performing grasped.

2.4 Summary

Table 2.1 summarizes all the information gathered about the techniques used by the previous researches which involved articles, review paper and conference papers for the five previous years. It categorized the image acquisition, image processing and image analysis techniques applied. As illustrated in Table 2.1, only two types of imaging device applied in the previous research. As example, it applied the basic camera [4]–[7], [15] and stereo camera [3], [7]–[12], [14], [16]. As mentioned in the objectives, this study needs to design and develop a planar gripper. So that, the gripper is only involves in the space which is two dimensional and not three dimensional. The suitable imaging device that will be implemented in this study is basic camera.

Furthermore, this study cannot be completed without image processing and image analysis technique. The implementation of the techniques similar as the previous researches used which is geometric feature extraction is not involved because it is related to three dimensional [3], [6], [9], [16]. For this study, OpenCV-Python software will be used as a platform to carry out the machine vision system process. This software was designed with many basic techniques that can be implemented such as global threshold, segmentation, Otsu method, canny detector, morphological, SVM and many more.

Table 2.1: Summary of Vision System In Previous Researches

#	Source	Image Acquisition	Image Processing	Image Analysis
[3]	Danny Eizicovits et al.	Stereo camera & point cloud	Boundary Box & geometric feature extraction	Mapping relationship
[4]	Rahul Kumar et al.	Basic camera	Otsu's method, Canny detector, morphological	Neural network
[5]	Matheus F. Reis et al.	Basic camera	-	Visual servoing
[6]	Vincenzo Lippiello et	Basic camera	Global threshold & geometric feature	-

	al.		extraction	
[7]	Yuanshen Zhao et al.	Basic camera, stereo camera, histogram equalization & etc.	Otsu's method & Canny detector	Support Vector Machine, Fuzzy clusterring & neural network
[8]	P. Aivaliotisa et al.	Stereo camera	-	Labelling, neural network & IBM Watson
[9]	Vincenzo Lippiello et al.	2 pairs of basic camera (stereo vision)	Geometric feature extraction	Kalman filter & BSP tree
[10]	Yingbai Hu et al.	Stereo camera & point cloud	Color-based segmentation	Support Vector Machine
[11]	Fei Chao et al.	Stereo camera & Novel filter	Global threshold	MATLAB
[12]	Fuchun Sun et al.	Stereo camera & histogram equalization	Global threshold	Neural network
[13]	Alex Vásquez et al.	-	Global threshold & Sobel filter	Gazebo simulator
[14]	Di Guo et al.	Stereo camera	Global threshold	IoU threshold & labelling
[15]	Lu Chen et al.	Basic camera & histogram equalization	Boundary Box	IoU threshold & neural network
[16]	Chuankai Liu et al.	Point cloud & 3D-filtering	Geometric feature extraction	Mapping relationship
[17]	Fazel Farahmando et al.	Anti-aliasing	Global threshold & morphological	Labelling

As simplified by Table 2.2, the gripping mechanism applied in previous researches categorized into three categories which are number of fingers, degree of freedom and mechanisms. By referring Table 2.2, most of the previous studies used two fingers for their gripper design [3], [7]–[9], [17]. Actually, two fingers gripper is the simple design and easy to control. For this case, the gripper will be used electrical drive which is servomotor.

Then, Table 2.2 stated that most previous researches used 1-DOF [3], [11], [14], [17] and 2-DOF [5], [7], [8], [10], [16]. In this study, the 1-DOF fingers are not suitable for the application of grasping the object in caging position. The suitable numbers of degree of freedom for the fingers used are 2-DOF. Instead, the 2-DOF that will be implemented for the robotic fingers will use the inverse kinematic mechanism. It is for deriving the orientations of joints by using the position of the object contact points.

Table 2.2: Summary of Grasping Mechanism in Previous Researches

#	**Source**	**# of Fingers**	**Degree-of-Freedom**	**Mechanism**
[3]	Danny Eizicovits et al.	2-fingers	1-DOF	Forward kinematic (arm)
[4]	Rahul Kumar et al.	-	-	-
[5]	Matheus F. Reis et al.	3-fingers	2-DOF	-
[6]	Vincenzo Lippiello et al.	4-fingers	4-DOF	Forward kinematic
[7]	Yuanshen Zhao et al.	2-fingers	1-DOF	Clamp
[8]	P. Aivaliotisa et al.	2-fingers	2-DOF	Parallel
[9]	Vincenzo Lippiello et al.	2-fingers	2-DOF	Forward kinematic
[10]	Yingbai Hu et al.	5-fingers	2-DOF	Kinematic redundancy
[11]	Fei Chao et al.	4-fingers	1-DOF	4 bar link
[12]	Fuchun Sun et al.	3-fingers	3-DOF	Forward kinematic
[13]	Alex Vásquez et al.	5-fingers	3-DOF	Inverse kinematic
[14]	Di Guo et al.	3-fingers	1-DOF	Parallel
[15]	Lu Chen et al.	-	-	-
[16]	Chuankai Liu et al.	4-fingers	2-DOF	Parallel
[17]	Fazel Farahmando et al.	2-fingers	1-DOF	Inverse kinematic (arm)

CHAPTER 3

METHODOLOGY

3.1 Overview

This section states all the processes, methods and procedures will be taken by student in this project. It shows the process has been through by the student in this Final Year Project (FYP). This section will explain the flow that need to be taken by the students to complete their FYP. Moreover, it also elaborates on how the project will be finished. The procedures that will be performed also describe in this section.

3.2 General Project Flowchart

Proposed methodology The FYP program is specified for final year students. It can be classified to two parts which are Final Year Project 1 (FYP 1) and Final Year Project 2 (FYP 2), both are proceed on the first and second semester in the final year of degree. In FYP 1, one of the important outputs is the information gathering get from the previous researches. It also carried out the problems that need to be solved by student. Other than that, FYP 1 determined the overall project objectives and setups to solve the problems. For easiest way to understand, students need to create a proposal for the project in FYP 1. The proposal needs to be presented in front of chosen panels to decide an approval for the project.

While in FYP 2, student needs to carry out the methods for problem solver and perform the experiments. It will be fully explained on how student will obtain the results. Then, students need to finish up the report writing and submit it. The result obtained need to be presented again. It is for the panels evaluating to verify that results achieve the objectives or not. Figure 3.1 shows the flowcharts need to be followed by student during the FYP 1 and FYP 2 program respectively.

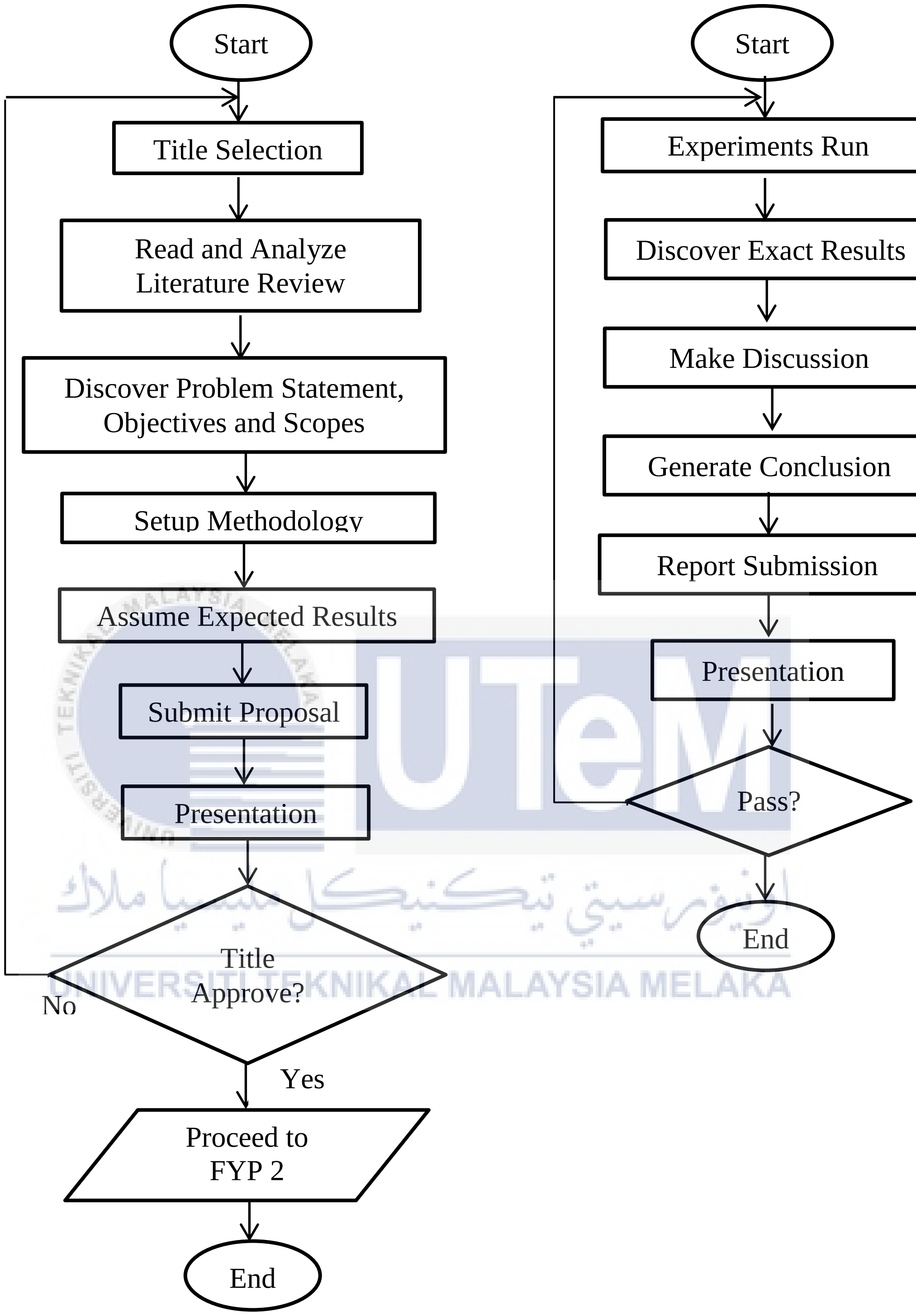


Figure 3.1: FYP 1 and FYP 2 Flowcharts

3.3 Specific Project Flowchart

This section elaborates the overall progress specified for the project. First of all, the design for the planar gripper needs to be constructed. It follows by the angular motion of the gripper movement. Then, it has the hardware and the software that needs to develop. Finally, carry out the real-implementation test for the product and run the experiments which are reliability test to determine the accuracy in angle analysis. Table 3.1 states all the tasks need to carried out for achieving the objectives. Shown in Figure 3.2 and Figure 3.3, the specific progress for the project illustrates by the Gantt chart and the flowchart. Last but not least, Figure 3.4 demonstrates the ways how the product for this project works in flowchart.

Table 3.1: Tasks Representing In the Objectives

Objective	Task	Description
1	1	Design gripper
	2	Determine initial position of angular motion
	3	Determine desire position of angular motion
2	4	Develop mechanical hardware
	5	Experimental setup
	6	Develop program for software
	7	Full process test run
3	8	Determine possible contact points
	9	Comparison pixel and real
	10	Reliability analysis

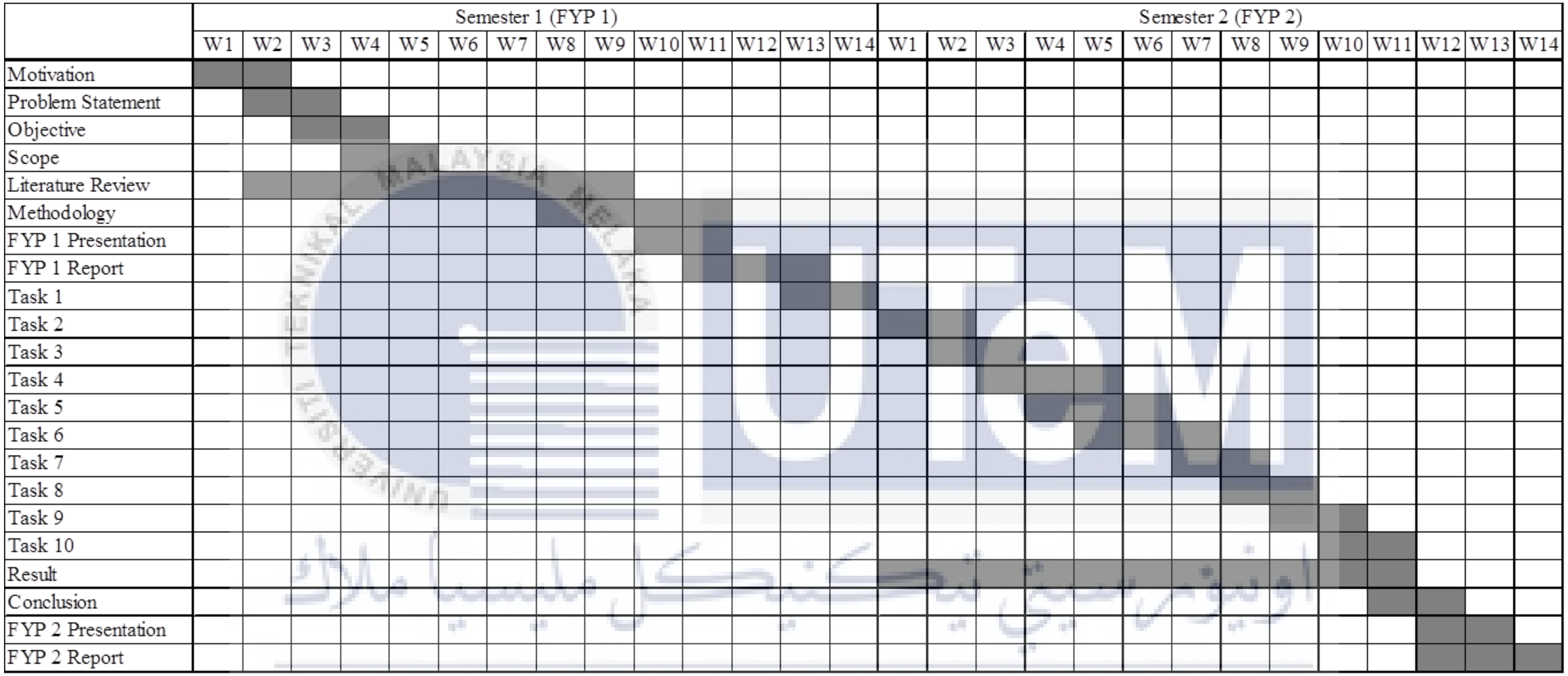


Figure 3.2: Gantt chart For the Final Year Project Program

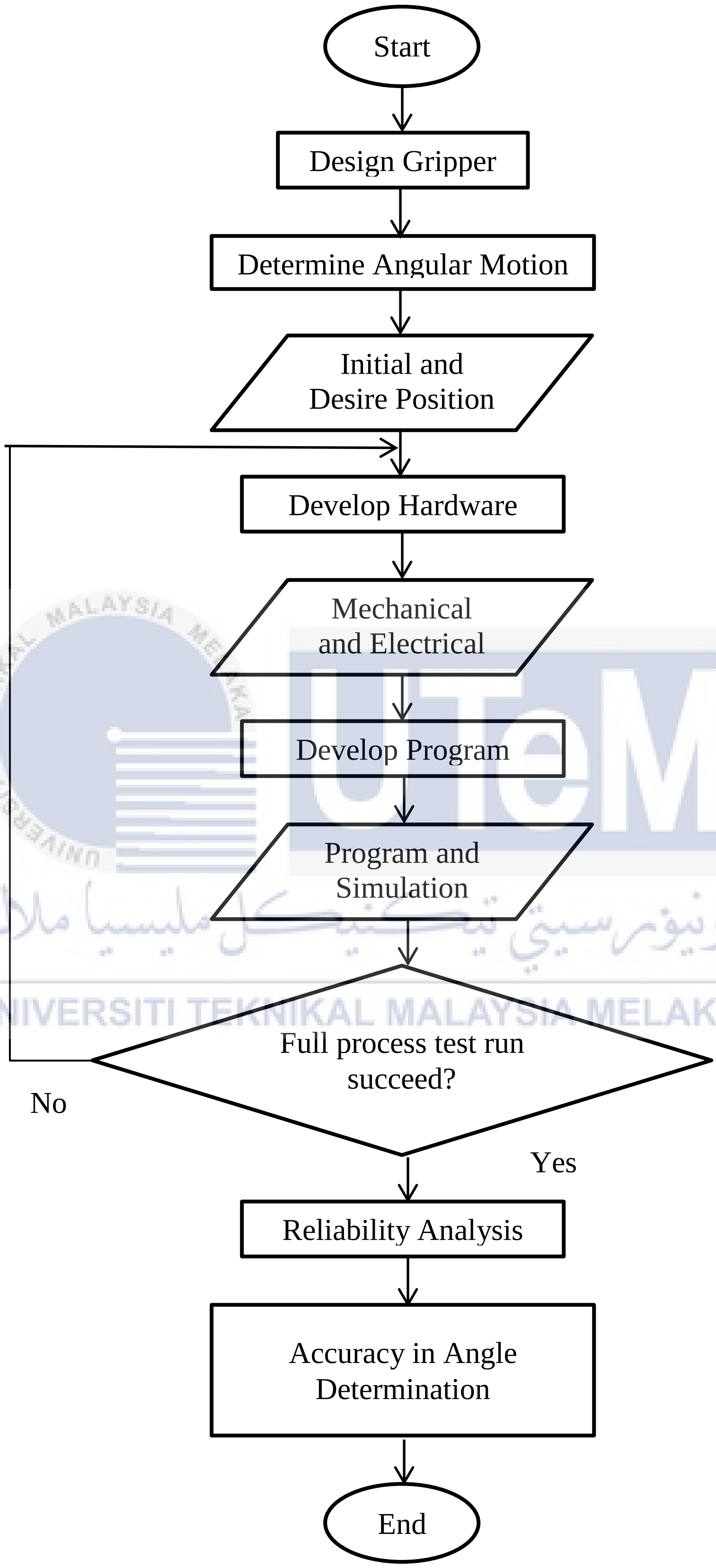


Figure 3.3: The Specific Project Flowchart

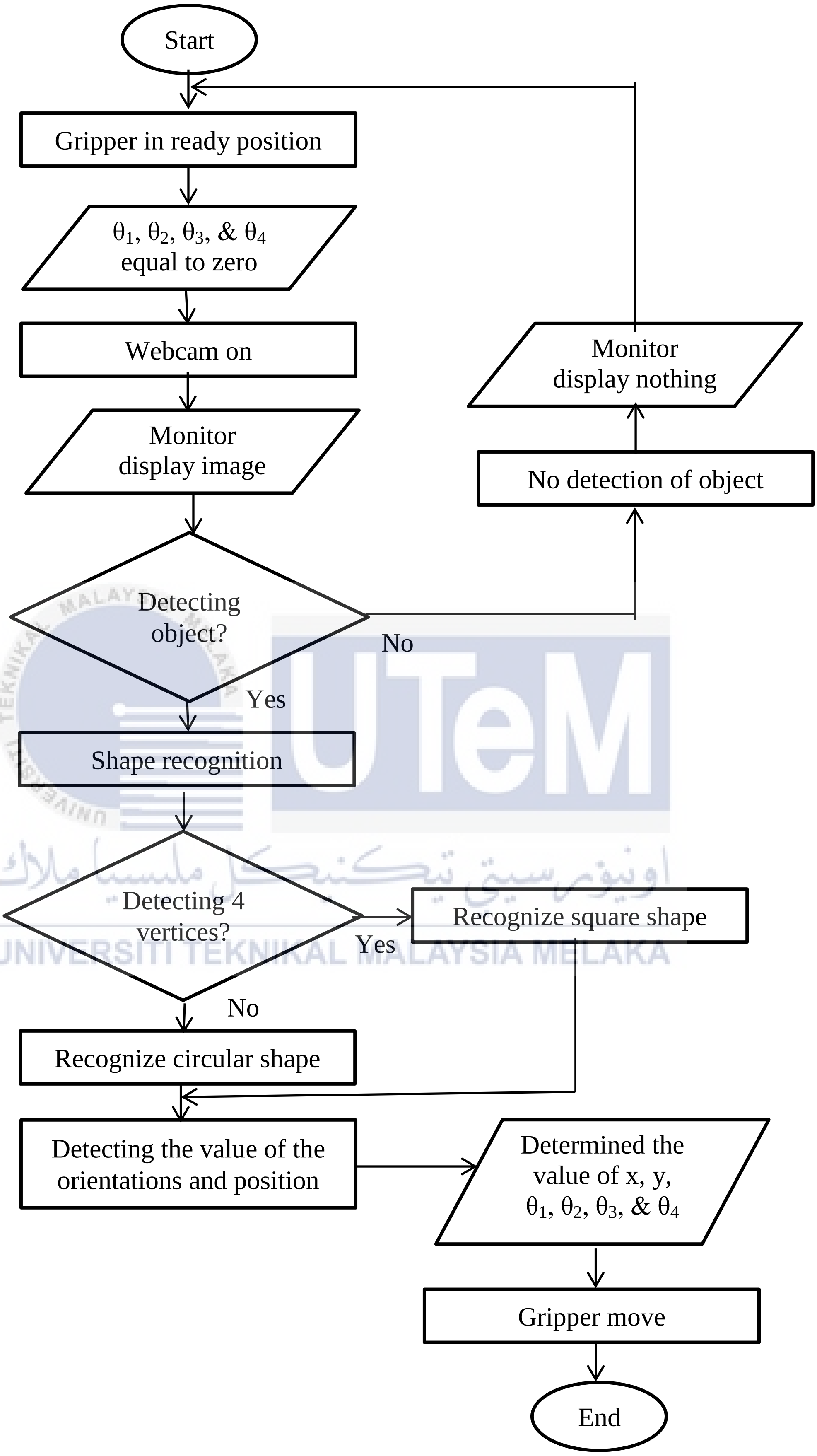


Figure 3.4: The Program Flowchart

3.4 Designation

In this section, it will be divided in three processes. At first, it is about the design processes take place. Secondly, the determination about the ready position set up of the gripper. Lastly, the descriptions on how the desire positions to grip object occurred.

3.4.1 Design Gripper

First of all, design the gripper started from two dimensional constructions by sketching and drafting. The AutoCAD software is used for completing the draft design. The design considered the requirement which is the fingers implemented 2 DOF. The final draft design decided and then three dimensional modeling takes place using SolidWorks software. The 3D modeling of gripper included the links, servomotors and the circle and square shape which are the objects that need to be gripped. 50.27 cm^2 and 72.25 cm^2 are the area of the circle and square object respectively. Figure 3.5 shows the logo of the software used for the process. Next, all of the parts assembled to become a complete gripper. The 3D model drawing file saved in SLT format.

Figure 3.5: AutoCAD and SolidWorks Software Logo

3.4.2 Initial Position

The gripper at the first place must be set up. It is in the ready position or initial position to grip the targeted objects. All of the orientation for each joint set to be at zero degree. The origin between the two moveable fingers and the location of the moveable fingers base considered as well as the reference point that locate the centroid of objects. All of the points involve in Cartesian space need to be measured and recorded.

After that, the contact points of the objects that will be touched by the one of the finger need to be assumed. Consider the positions of x-axis and y-axis of each 5° of the objects outer surface from 0° to 45° of first quadrant for the circle object and 315° to 360° of the last quadrant for square object. The parameters involved for Finger 2 are unnecessary to find because it use the same output. Figure 3.6 and Figure 3.7 demonstrate all the require position in Cartesian space for the circle and square object gripping respectively. The tables created to record all the data obtained.

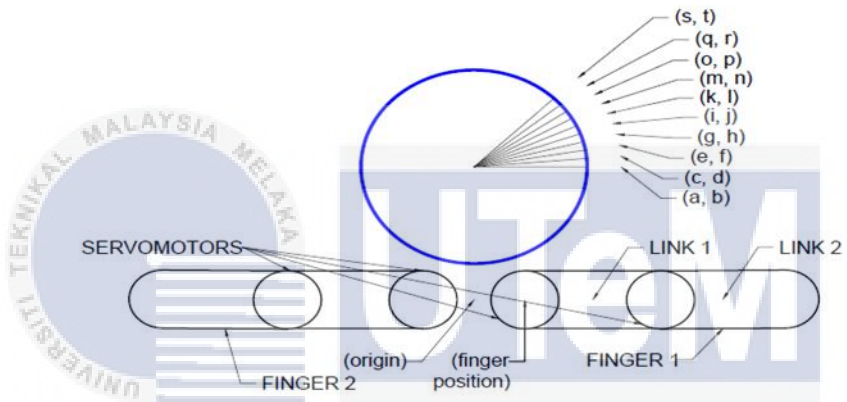


Figure 3.6: The Required Positions for Gripping the Circle Object

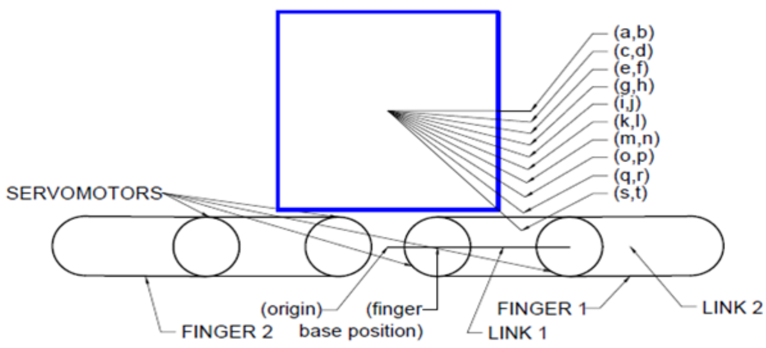


Figure 3.7: The Required Positions for Gripping the Square Object

3.4.3 Desire Position

At last, the desire position that the moveable fingers will be moved by using inverse kinematic calculation is determined. The Cartesian space converted to Joint space by using the position determined from previous step to find the orientations of the Finger 1 joints. Figure 3.8 illustrates the parameters involved the inverse kinematic equation.

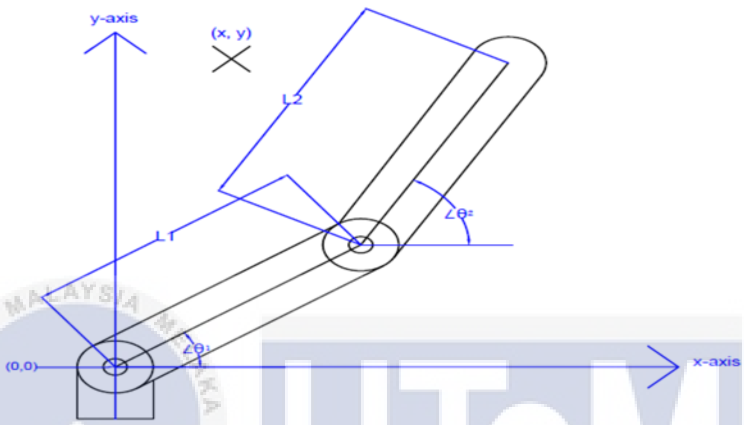


Figure 3.8: The Parameters Involve in Inverse Kinematic Equation

L_1 and L_2 are the length of links. θ_1 and θ_2 are the orientation of the joints. Then, (0, 0) is the origin which is the base of the robotic manipulator and (x, y) is the position of the contact points that will be reached by the robot. The length of links measured and make sure the position of the contact point refers to the coordination of manipulator. The equation (3-1), (3-2), (3-3), and (3-4) need to compute using the value acquired.

(3-1)

Orientation Link 2, $\boldsymbol{\theta_2} = \mathbf{cos}^{-1}\left(\frac{x^2+y^2-{l_1}^2-{l_2}^2}{2l_1l_2}\right)$

(3-2)

Constant 1, $k_1 = l_1 + l_2\cos\theta_2$

(3-3)

Constant 2, $k_2 = l_2\sin\theta_2$

(3-4)

Orientation Link 1, $\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{k_2}{k_1}\right)$

All of this formula applied to find the orientations of the possible contact points assume for gripping the circle and square object. The data acquired need to be recorded. Then, the movements of the finger are analyzed by using the orientations obtained. Each of the movement needs to be stated whether it is succeed or fail to grip the objects.

Additionally, another approach can be used to determine the contact points. Logically, the surface of links can touch the object surface. The lengths of both links are divided into 10 divisions. The circumference of the circle and the surface of the square also need to be divided. Then, calculation needs to carry out for the first link at first because the orientation acquired involved in the calculation for the second link. Two contact points can determine the orientation of the link 1 and link 2 respectively. Equation of (**3-5**), (**3-6**), (**3-7**), and (**3-8**) used to find the possible orientations.

For first link,

$$x_1 = l_1 \cos\theta_1 \quad (3\text{-}5)$$

$$y_1 = l_2 \sin\theta_1 \quad (3\text{-}6)$$

For second link,

$$x_2 = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (3\text{-}7)$$

$$y_2 = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (3\text{-}8)$$

3.5 Development of Hardware

This section elaborates on assemble of mechanical and electronic hardware that equipped in the study. It can be categorized into two parts which are mechanical design and electronic connection. Actually, these processes are part of the first objective as well as the second objective but more rely on accomplishing the second objective. It is also the procedure need to be followed for the experimental setup.

3.5.1 Mechanical

First of all, the gripper links that have been drawn in previous procedure need to be prepared. The links of the fingers were built by using 3-dimensional printing. The Rapid Prototyping Machine called as 3D printer will read the drawing only in the SLT file format. This procedure required 3D printer and acrylonitrile butadiene styrene (ABS) as a material to develop the hardware designed. Figure 3.9 shows the 3D printer printing the model after uploaded the SLT format files.

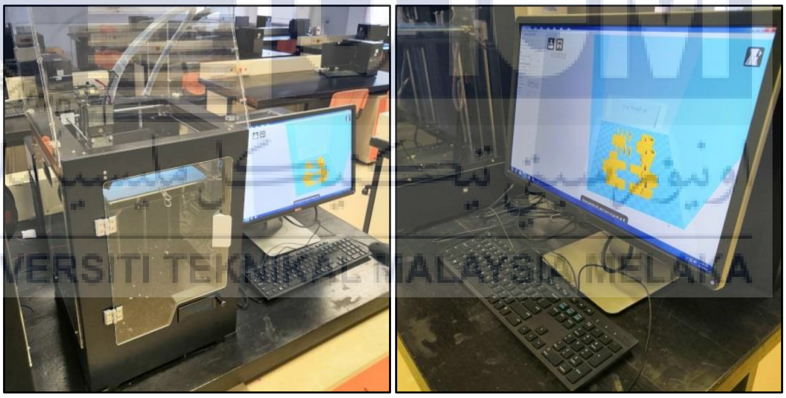


Figure 3.9: 3D Printer Printing the Model

After that, the servomotor installed by connecting the links printed. Each finger required two servomotors and two links. The controller implemented is placed in the box. The first joint of each finger mounted to the box acts as the base. Refer to the position of base on Cartesian plane drafted before.

Meanwhile, the setup needs to be prepared as the workspace for the vision system. It is consists of the 50 cm height of webcam holder for holding the webcam at the correct position, the A3 paper size of grid background snapped in the picture and the proper board to locate all of this equipment. Figure 3.10 represents the workspace for this project. Next, all the parameter which are the actual origin of the Link 1 as well as the actual lengths of Link 1 and Link 2 have been measured and recorded.

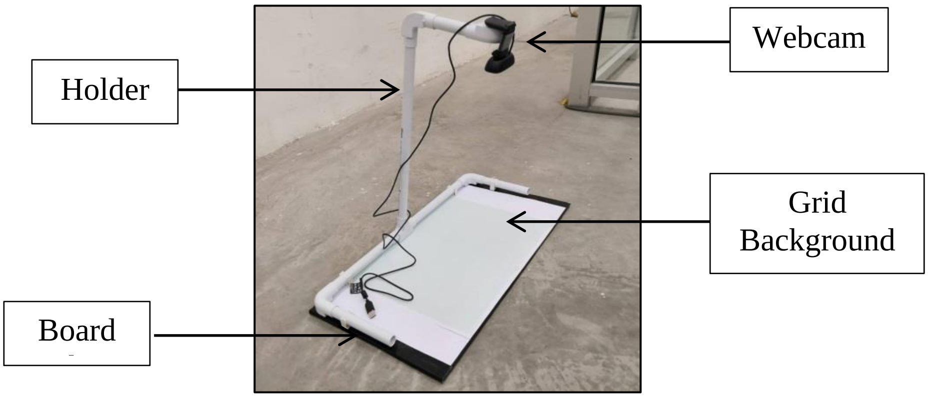


Figure 3.10: The Workspace

3.5.2 Electronic Connection

For the circuit connection, it involves the Raspberry Pi and Arduino controller. Figure 3.11 shows the Raspberry Pi and Arduino control module. Raspberry Pi acts as a platform for the vision system while Arduino acts as a gripper controller to control the two moveable fingers. This connection also should involve a webcam, monitor, keyboard, mouse and 4 units of servomotors.

Figure 3.11: Raspberry Pi and Arduino Controller Module

For the vision system, a webcam connected to Raspberry Pi as an input need to be chosen wisely. For this case, Logitech Webcam C110 has been used. Plus, the inputs also consist of keyboard and mouse. A monitor connected as output to display an image produced. Figure 3.12 illustrates the connection of the electronics equipment for the vision system implemented.

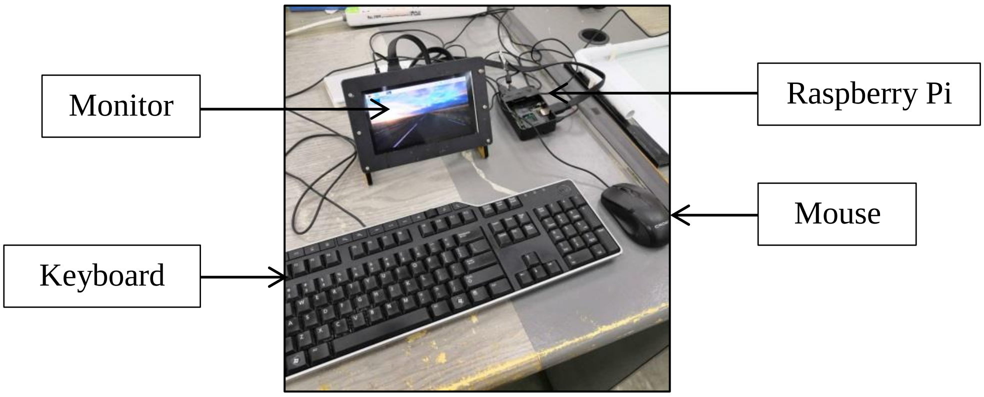


Figure 3.12: The Connection of the Electronics Equipment for Vision System

3.6 Development of Program

This section describes about the program involved for machine vision system and gripping mechanism applied. The techniques that will be used for recognizing and gripping the shape represents in coding. The source code will be divided into two which for the OpenCV-Python and Arduino as platforms for running the vision system and grip mechanism respectively. Figure 3.13 represents the entire software tools implement in this study.

OpenCV python ARDUINO

Figure 3.13: All Software Tools Implemented

As mentioned before, machine vision system applied OpenCV-Python software as a platform. There are several techniques used to recognize the shape. At first, a function developed to run shape detector by determine the vertices of the object. For this study, it only consists two shapes either it circle or square. The amounts of vertices are recognized by using a contour. If the object has four vertices it will recognize square shape. Else, it will recognize circle shape.

Next, the function created for the program. The acquire image captured by webcam. It converted to gray-scale image. Threshold used to convert the gray scale to binary image. The high frequency noise is reduced by blurring at the same time it removed the shadow and the background. The function helped the object to be highlighted for being easy to differentiate the object with the unnecessary element existed. It shown in the Figure 3.14, the acquired image of two objects with different shape converted into the new image with the dark background.

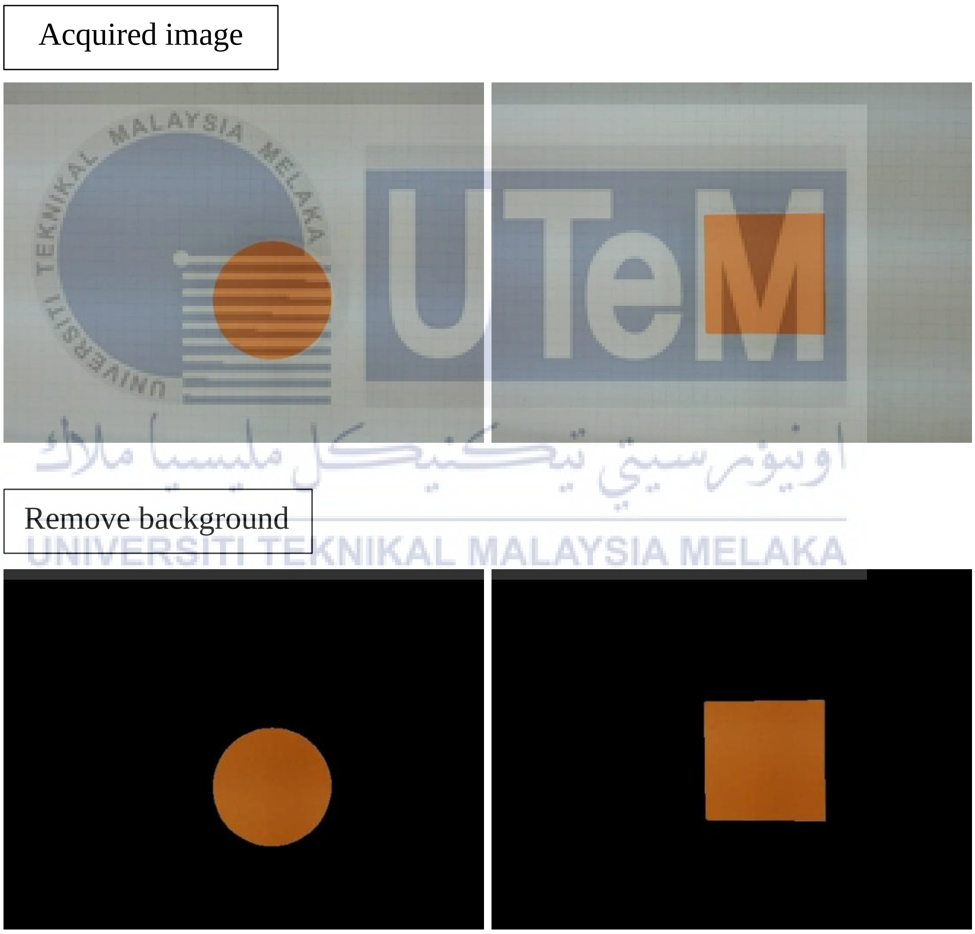


Figure 3.14: The Initial Image Converted To New Image with Dark Background

Next, edge detection and contour applied for detecting the outline of the object to recognize the shape. Image moments computed for the region detected. Image moments technique helps to determine the length, radius, centroid and etc.

The shape detection function called to recognize the shape of the object either it is circle or square. The contour is shown and the label is written on the object. Last, the result displayed on the monitor. All of these functions are written by using OpenCV-Python software.

The fingers movement controlled by the program is written in Arduino IDE software. The output of the vision system produced as the input for Arduino. The program written by set the degree of the rotation of servomotors to zero as the gripper ready position. The function added to determine the degree of the rotation of servomotors to reach the contact point. The orientation obtained in previous section implemented to grip each shape of object. The results obtained are verified and evaluated.

3.7 Experiments

This section explains on the methods implement to carry out the experiments. Three experiments conduct namely full image process test run, comparison between pixels parameters and real parameters and reliability analysis on the angle determination. The procedures involved in each experiment will be fully described.

3.7.1 Image Process Test Run

First experiment conducted to run the full process of the vision system designed. This test run followed the previous procedures. Initially, the purpose of this experiment is just to identify the shape of the object either it is circle or square. So, the experiment involved two objects with different shapes. The result obtained need to be illustrated. Other than that, the second experiment conducted with only involved a circular shape with three different sizes which are 50.27cm2, 28.27cm2 and 12.57cm2 for Object 1, Object 2 and Object 3 respectively as shown in Figure 3.15. The acquired images for all of the objects are taken with the webcam. All of it located on five locations on the board which are center, top left, bottom left, top right and bottom right. Then, the unnecessary background converted to dark background. The contour applied to help the vertices of the object been recognized. This experiment determined the effectiveness of the vision system applied on the similar shape in different sizes and locations.

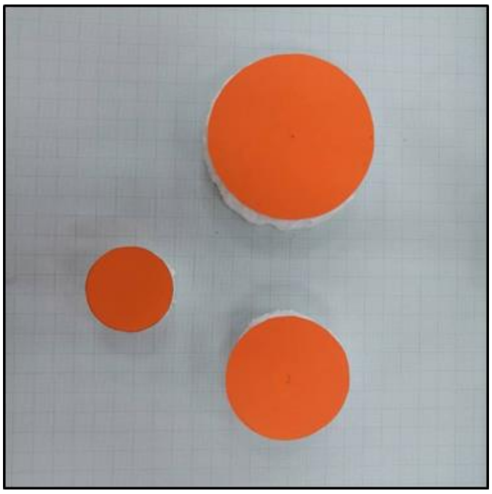


Figure 3.15: Three Objects with Similar Shape and Different Sizes

3.7.2 Comparison between Actual Parameter and Pixels Parameter

Two investigations conducted for comparing the real value and pixel value which used two objects with different shapes per analysis. It used the images obtained from the previous shape recognition analysis. It also conducted to determine the accuracy of the vision system performed. Each pixel has it value that determines the position of object's edge in image.

The pixel values compared with the actual location of the objects on the real coordination. Centroid coordinate for both shapes of each object are recorded. The perimeter involved in circle shape such as radius also been recorded and same goes to the square shape perimeters such length and width. All of these values in pixels are compared to the real values of the objects. The values of the real coordinates obtained from the grid that was set as the background. The values represented as x-axis and y-axis in Scatter chart which are vertically and horizontally on the workspace and also the image. The object located on the middle location of the board.

Other than that, the angular movements of the gripper perform based on the possible contact points by using the inverse kinematic mathematical approach. The

10 contact points are manually calculated using the coordinates of the parameter for each shapes in pixel values and real coordinate. The pixels data are recorded and compared to the real coordinates. The accuracy of each contact point and its average can be determined.

3.7.3 Reliability Analysis on Angle Determination

The orientations applied by the fingers been analyzed for the purpose of the final experiment. As described before, the orientations are set using the value determined by previous method. This experiment used these value matched to the angular rotation of the servomotors performed. The system runs in several times. The angular rotation of each servomotor in each joint of Finger 1 is measured and recorded. The error determined by comparing the values with theoretical value. The accuracy of the system can be analyzed in determine these orientations. The analysis made by comparing the angular movement for orientation obtained from both approaches.

CHAPTER 4

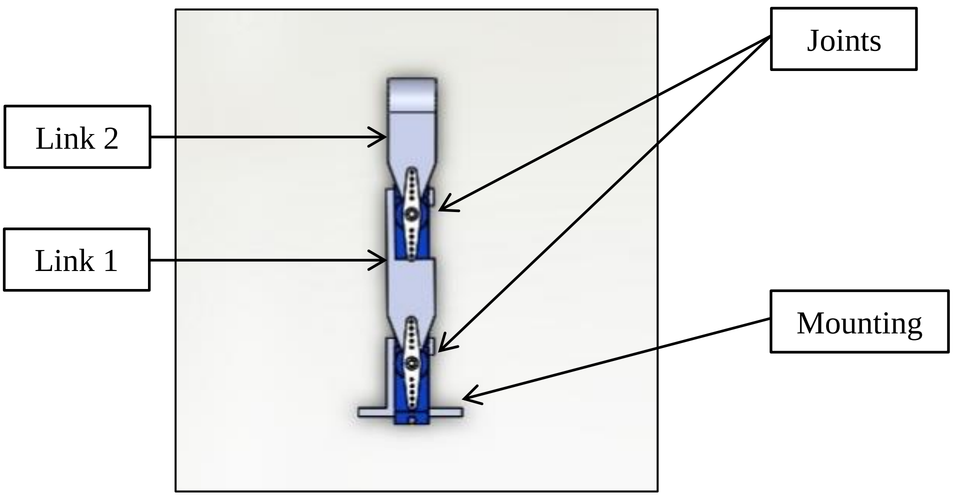
RESULTS AND DISCUSSIONS

4.1 Overview

This section represents the results acquired and the discussion that can be made from the results. The results involved the consequences in every element from each stage in the processes. Technically, the discussion made from several approaches which are analysis, synthesis and evaluation.

4.2 3D Modeling

This design followed the requirements needed which have two links and two joints for each finger as well as the two fingers mount on the exact position as a base. Figure 4.1 illustrates the design for one of the fingers drawn by using SolidWorks software tool. In this illustration, the design included the servomotors that were applied in this project. Figure 4.2 represents the model that was printed by 3D Printer. The final model complete assembled have 5.2 cm and 4.5 cm lengths for Link 1 and Link 2 respectively.



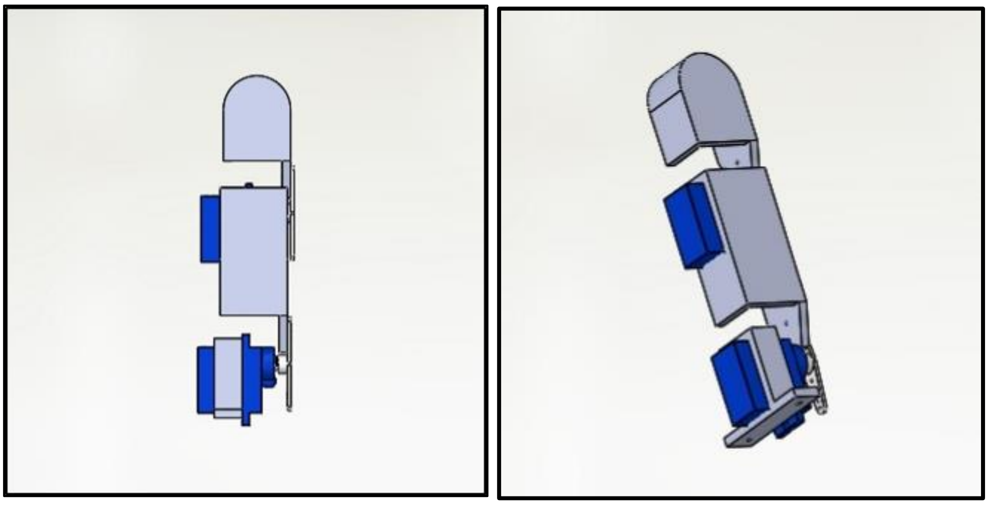


Figure 4.1: The Design of Two Fingers Gripper with 2-DOF

Figure 4.2: The Model Printed By 3D Printer

4.3 Image Process Test Run

This section explained two subsections consist of the result obtained from the program to recognize the different shapes and its effectiveness.

4.3.1 Shape Recognition

The result appeared on the monitor determine either the target object is circle or square. As mentioned in methodology, the program set to identify the value of vertices. Four vertices for square and more than four vertices is circle. Figure 4.3 and

Figure 4.4 illustrates the result for determined the object that has circle and square shape respectively.

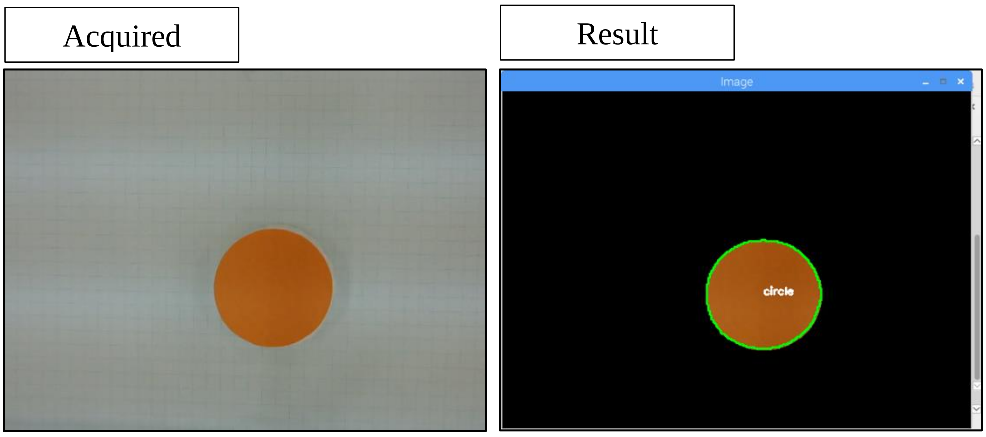


Figure 4.3: The Result Determine the Object That is Circle Shape

Figure 4.4: The Result Determine the Object That is Square Shape

The experiments conducted to test the successful of the program to recognize the shape. It used two objects with different shapes. As the results, Figure 4.3 shows the result achieved which is the program is success to recognize circle and Figure 4.4 shows the program accomplished to recognize square. In addition, the program also helped to illustrate the edge boundary and write the text labeled as the shape of the object.

4.3.2 Vision System Effectiveness

In this experiment, three objects with similar shape with different sizes were tested with the vision system used. It is also used the same techniques. The image acquired for these objects are located at five different places. So that, total image that have been processed are 15 images.

Figure 4.5 represents the acquire images for Object 1 in all desired locations and Figure 4.6 demonstrated the results. It shown that the shape of the objects at all locations has been successfully identified. Table 4.1 shows the recorded data that related to the results which are the actual position of the object, the position of the Object 1 in acquired image, and the result indicated on the monitor. Figure 4.7, Figure 4.8 and Table 4.2 represent the related result for Object 2 as well as Figure 4.9, Figure 4.10 and Table 4.3 for Object 3.

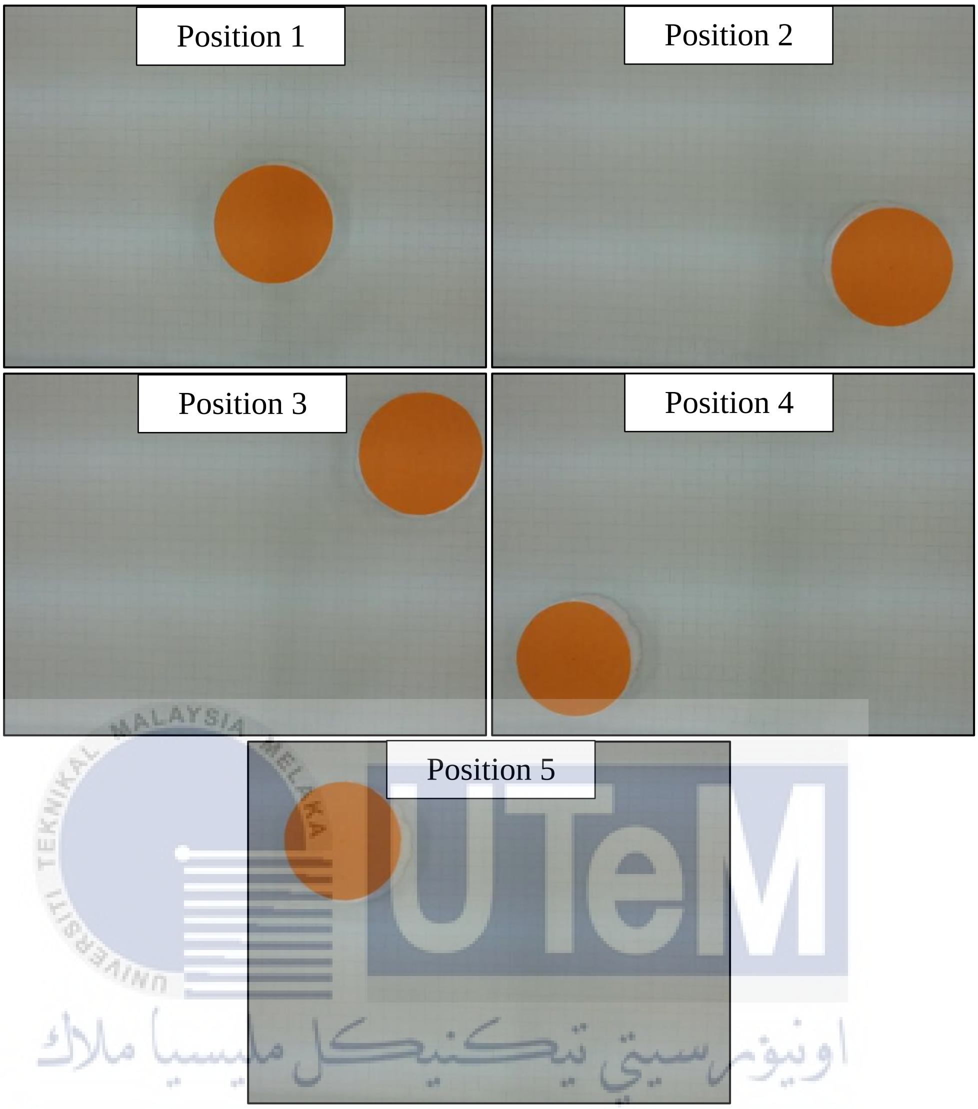


Figure 4.5: The Acquired Image for Object 1 at All Desired Position

In Figure 4.5, it shows all five acquired images. All these five acquired images are the image of Object 1. Object 1 is the large size between three objects represented in this experiment. Each acquired image captured the object at five different positions. Position 1 is the middle position for the object located in real situation. The acquired image also represented the object is at the middle. Meanwhile, the object is detected on the bottom right in the acquired image when the object is located at top left in real situation represented in Position 2. Then for Position 3, the object was at bottom left in the real situation but in the acquired image it was represented at top right. Similar to Position 4 and Position 5, the real locations of the object are at top right and bottom right but in the acquired images the objects change its locations to the bottom left and top left respectively.

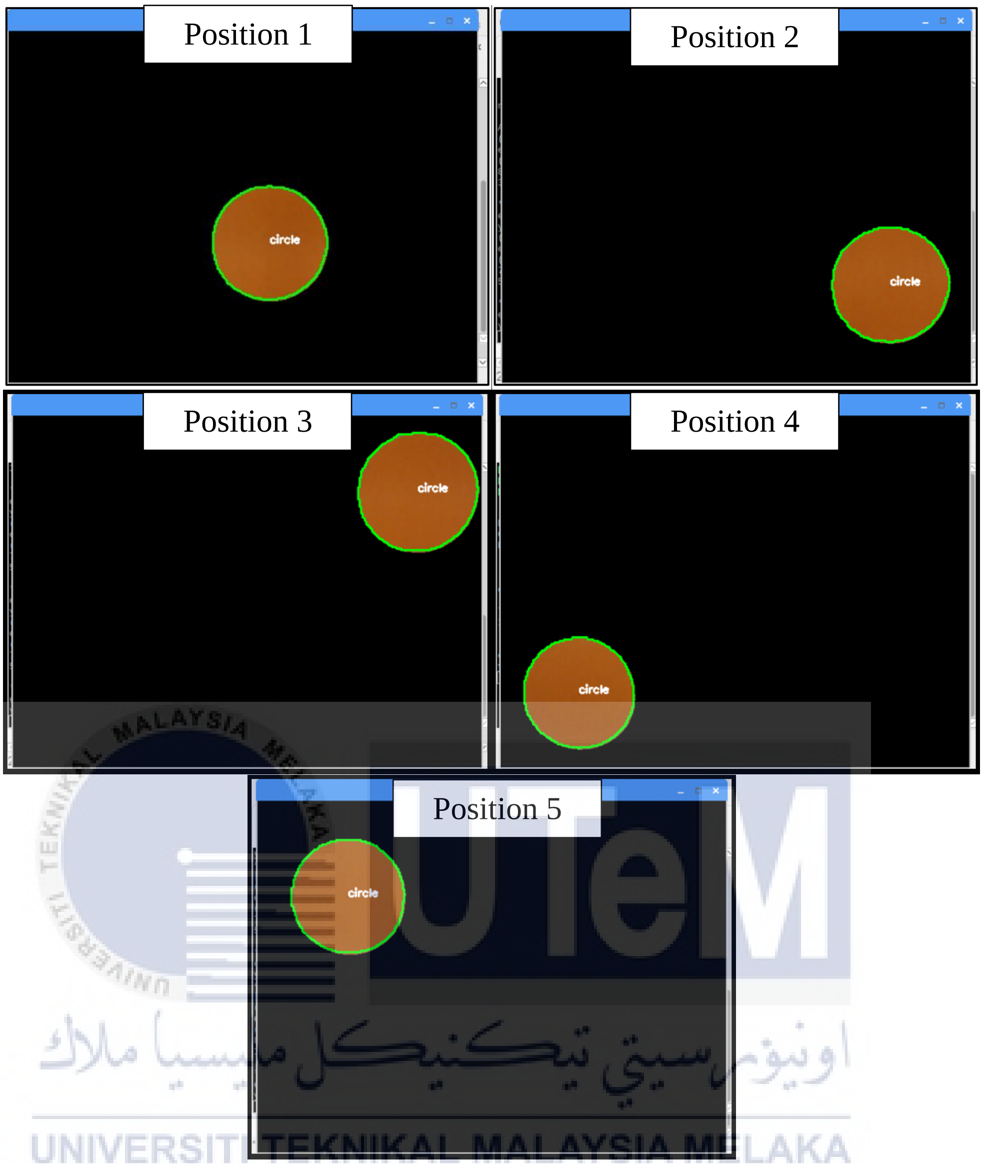


Figure 4.6: The Result for Object 1 at All Desired Position

As the results shown in Figure 4.6, all of the acquired images are succeed in shape recognition. The displays revealed all the results are correct which the object in all acquired images is circle. The program is success to detect the object and it also labeled the object as circle.

Table 4.1: The Data Obtained for Object 1

Position	**Object Position in Real**	**Result**		
		Shape	**Object Position in Image**	**Status**
1	Middle	Circle	Middle	Correct
2	Top Left	Circle	Bottom Right	Correct
3	Bottom Left	Circle	Top Right	Correct
4	Top Right	Circle	Bottom Left	Correct
5	Bottom Right	Circle	Top Left	Correct

From the Table 4.1, it stated Object 1 at all position tested are correct in recognizing the shape. It proves the program is 100% successful for detecting the shape in large size. It also shown the changes of object's locations have been made in real position to acquired image.

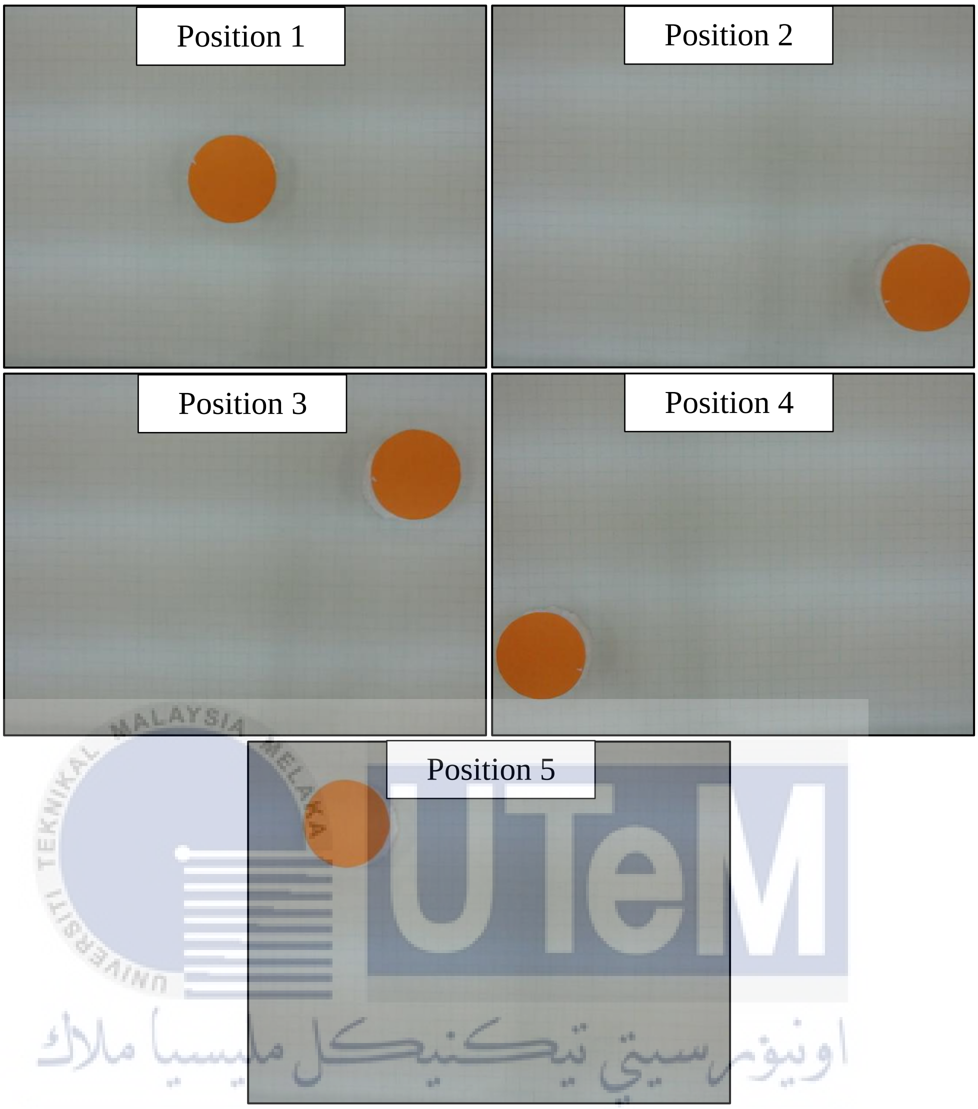


Figure 4.7: The Acquired Image for Object 2 at All Desired Position

In Figure 4.7, it shows all five acquired images for Object 2. Object 2 is the average size between three objects represented. Each acquired image captured the object at five different positions. All positions in acquired images changed from the object's real positions same as mentioned previously for Object 1.

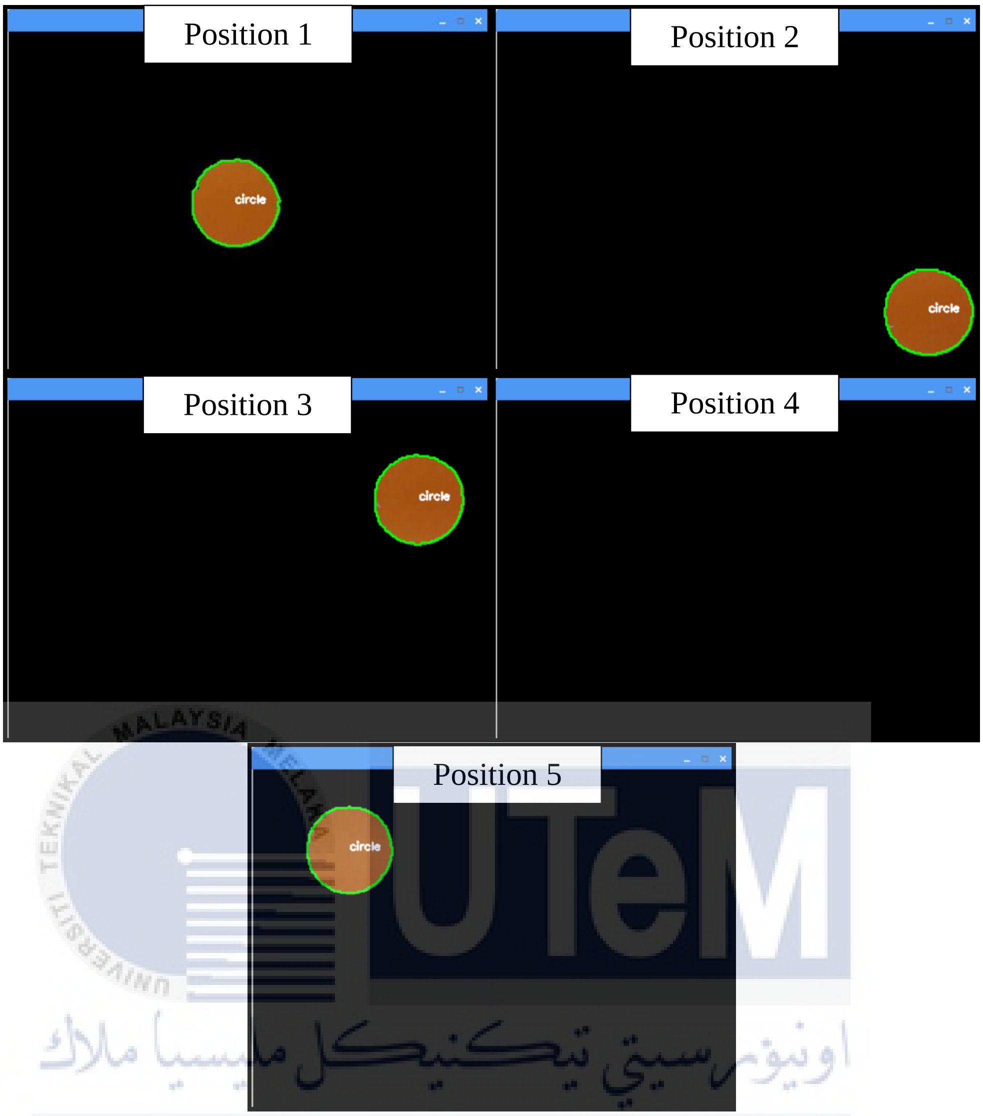


Figure 4.8: The Result for Object 2 at All Desired Position

As the results shown in Figure 4.8, all of the acquired images are succeed in shape recognition except for Position 4. The displays revealed all the results are correct which the object in all acquired images is circle except for Position 4. From the acquired image in Figure 4.7 for Position 4, there is the existence of shadow presented in the image. So that, the result for shape recognizing is fail. The object was disturbed to be highlighted then the program cannot recognize the circle.

Table 4.2: The Data Obtained for Object 2

Position	**Object Position in Real**	**Result**		
		Shape	**Object Position in Image**	**Status**
1	Middle	Circle	Middle	Correct
2	Top Left	Circle	Bottom Right	Correct
3	Bottom Left	Circle	Top Right	Correct
4	Top Right	*None*	*None*	Fail
5	Bottom Right	Circle	Top Left	Correct

From the Table 4.2, it stated Object 2 at all position tested are correct in recognizing the shape except for Position 4. The program was failed in detecting the shape and position of the Object 2 for Position 4. It proves the program is 80% successful for detecting the shape in average size. It also shown the changes of object’s locations have been made in real position to acquired image.

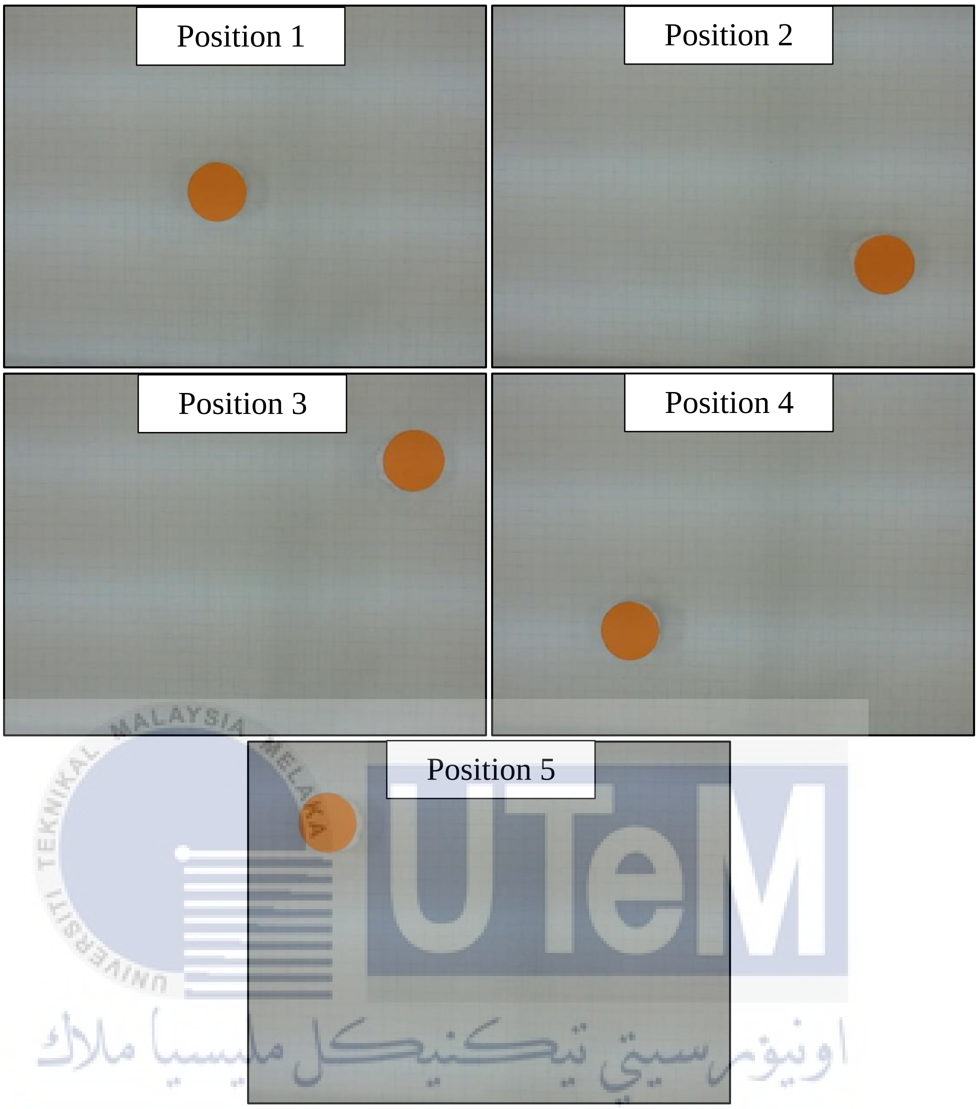


Figure 4.9: The Acquired Image for Object 3 at All Desired Position

In Figure 4.9, it shows all five acquired images for Object 3. Object 3 is the small size between three objects represented. Each acquired image captured the object at five different positions. All positions in acquired images changed from the object's real positions same as mentioned previously for Object 1 and Object 2.

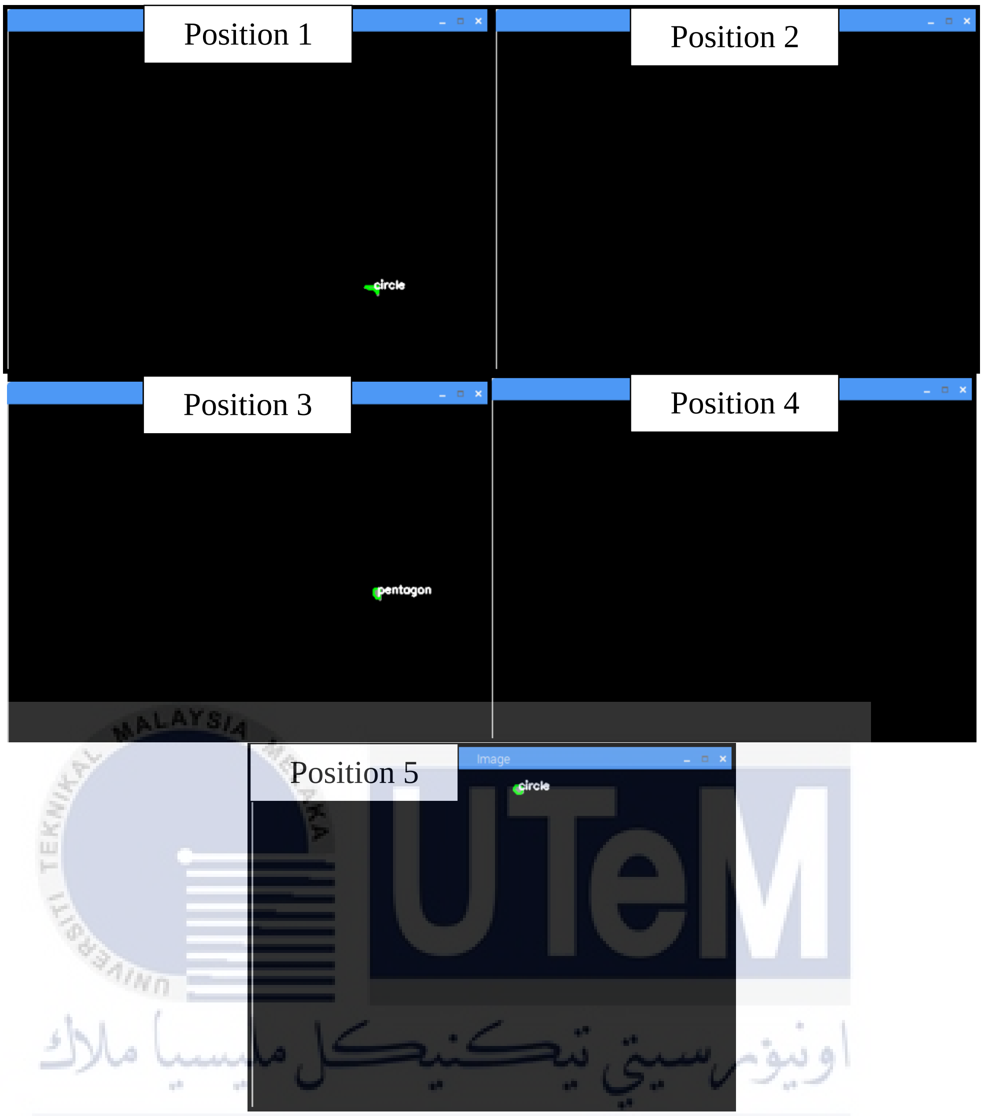


Figure 4.10: The Result for Object 3 at All Desired Position

As the results shown in Figure 4.10, all of the acquired images are failed in shape recognition. The displays revealed the results are correct for Position 1 and Position 5. It also shown the program cannot detect the object for Position 2 and Position 4. The result of shape detection was wrong for Position 3 which it recognized the shape was pentagon. The result for shape recognizing is fail due to the small size object was failed to be highlighted then the program cannot recognize the circle.

Table 4.3: The Data Obtained for Object 3

Position	**Object Position in Real**	**Result**		
		Shape	**Object Position in Image**	**Status**
1	Middle	Circle	Bottom Right	Fail
2	Top Left	*Fail*	*None*	Fail
3	Bottom Left	Pentagon	Bottom Right	Fail
4	Top Right	*None*	*None*	Fail
5	Bottom Right	Circle	Top	Fail

From the Table 4.3, it stated Object 3 at all position tested are failed in recognizing the shape. The program was failed in detecting the shape and position of the small size object. It also shown the changes of object's locations have been made in real positions to acquired images were not the same as the results in previous experiment for Object 1 and Object 2.

From the result, all five acquired images for Object 1 gave the correct required output which is the object is circle. Only four over five acquired images for Object 2 are correct and one of them not giving any result when it is at position the top right on the board. It is because there is an existence of shadow for the Object 2. The light intensity cannot fully cover up the object. Nevertheless, there is no output that is correct for Object 3. It is because the object is the smallest in size among the others. The camera used cannot highlight the small object as Object 3 appropriately.

4.4 Comparison between Real Parameter and Pixels Parameter

The first experiment for this section obtained 1,040 pixel values of circle's edge represented in image. The values obtained consist of the values of width and height. Figure 4.11 illustrates the Scatter chart edge boundary of the circle as the result. From the Scatter chart, the position of the object can conclude in the middle of the image same as the object in the image represented in Figure 4.3.

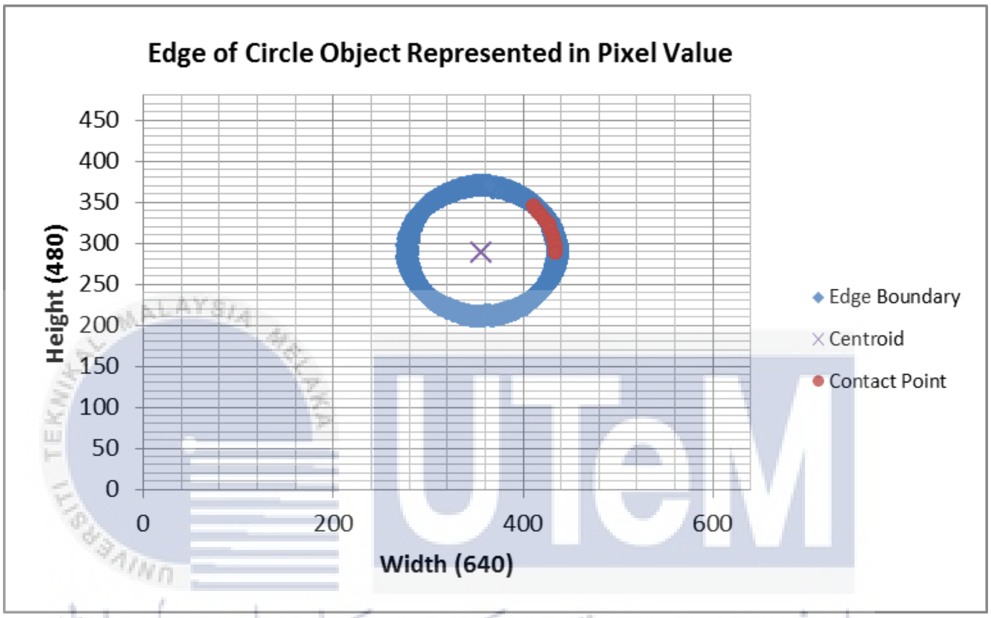


Figure 4.11: The Edge of Circle Object Represented in Pixel Value

Table 4.4 represented the require data of the circle parameter obtained in actual value and pixel value. The result can be analyzed as the image converted the real situation to 2.05 scales which also represents 2.05 pixels per millimeter.

Table 4.4: Actual and Pixel Values of the Circle Parameter

Parameter	Actual Value	Pixel Value	Ratio (pixels/mm)
Center (x-coordinate)	178	356	2
Center (y-coordinate)	152	289	2.11
Radius	40	78	2.05
		Average	2.05

Table 4.5 represented the actual and pixel values for the 10 possible contact points for the robotic finger to touch for gripping the circle object. From the data interpreted in the table, the average contact points accuracy which are 97.21% and 92.96% represented the horizontal coordinate and vertical coordinate respectively. The pixel value of the possible contact points were converted to real value using 2.05 pixels per millimeter obtained from the ratio calculated.

Table 4.5: Actual and Pixel Values of the Circle Contact Points

Contact Point	**Angle**	**Actual Value (x, y)**		**Pixel Value (width, height)**		**Accuracy (horizontal, vertical)**	
1	45°	206.28	180.28	200.49	167.80	97.19%	93.08%
2	40°	208.64	177.71	202.93	165.37	97.26%	93.05%
3	35°	210.77	174.94	204.88	162.93	97.20%	93.13%
4	30°	212.64	172.00	206.83	160.00	97.27%	93.02%
5	25°	214.25	168.90	208.29	157.07	97.22%	93.00%
6	20°	215.59	165.68	209.27	154.15	97.07%	93.04%
7	15°	216.63	162.35	210.24	150.73	97.05%	92.84%
8	10°	217.39	158.95	211.22	147.80	97.16%	92.99%
9	5°	217.85	155.49	211.71	144.39	97.18%	92.86%
10	0°	218.00	152.00	211.71	140.98	97.11%	92.75%
					Average	97.21%	92.96%

Figure 4.12 illustrates all the possible contact points in both actual and pixels value for circle object. From both graphs, the patterns are similar. It indicated the 10 possible contact points that the Link 2 of robotic finger will be touched. All of the possible contact points were located on the first quadrant of the circle object.

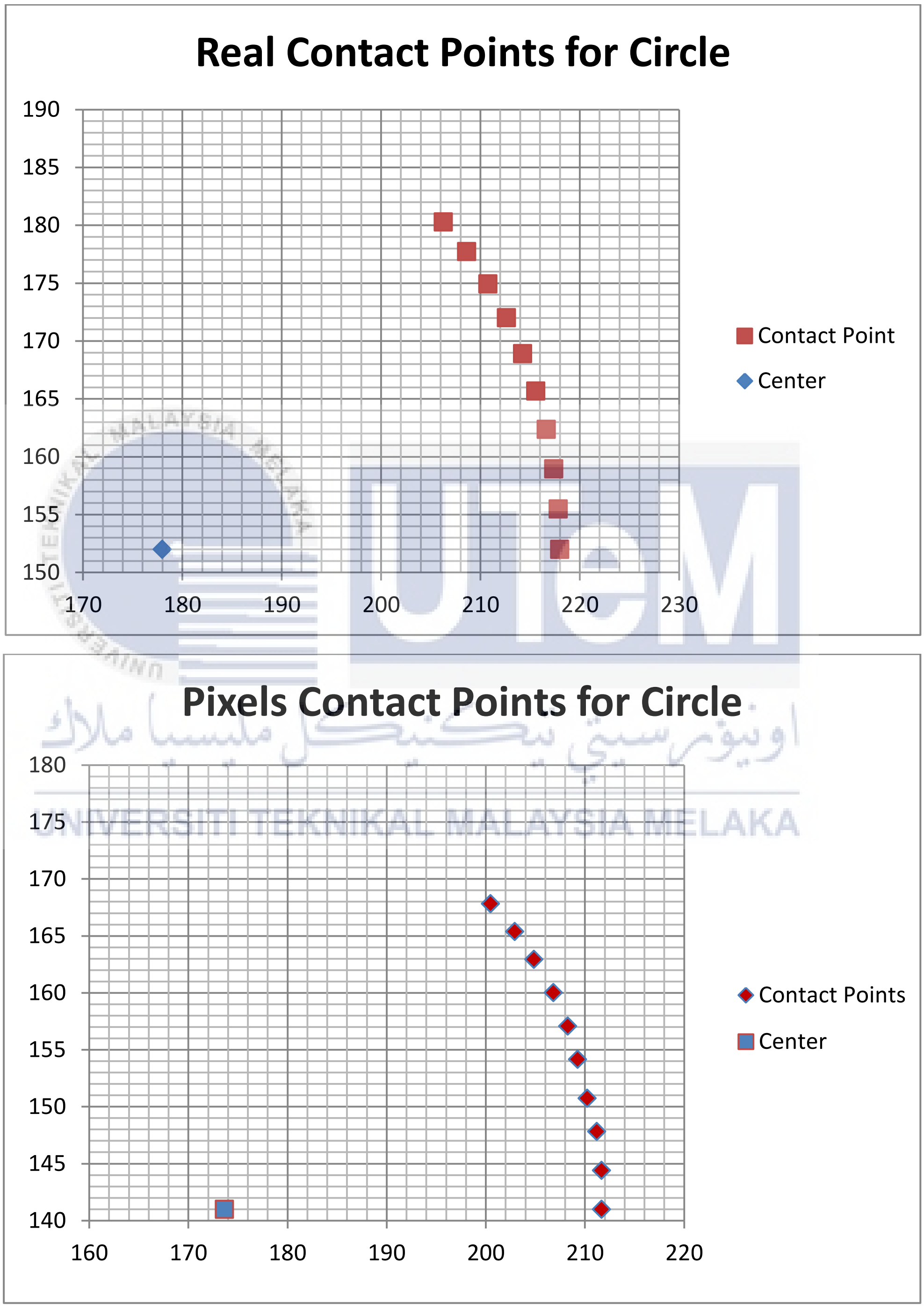


Figure 4.12: All Possible Contact Points in Actual and Pixels Value for Circle Object

Another experiment conducted for square shape object also used the image acquired in the previous shape recognition analysis. The result obtained by running the program is consists of 1,276 data of pixel values interpreted in the Scatter chart. Figure 4.13 shows the edge boundary of the square in the Scatter chart. It also showed the position of the square at the middle same as in the acquired image.

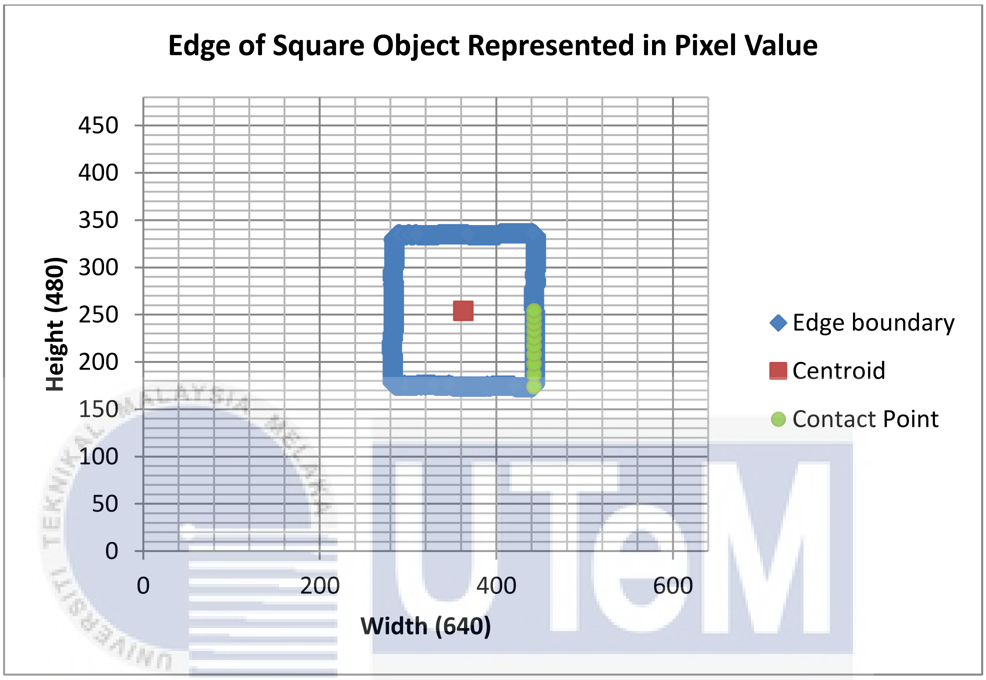


Figure 4.13: The Edge of Square Object Represented in Pixel Value

Table 4.6 represented the require data of the square parameter obtained in actual value and pixel value. The average of accuracy obtained for the parameters can be analyzed as the image converted to real values to 2.0 pixels per millimeter.

Table 4.6: Actual and Pixel Values of the Square Parameter

Parameter	**Actual Value**	**Pixel Value**	**Ratio (pixels/mm)**
Centroid (x-coordinate)	180	363	2.02
Centroid (y-coordinate)	130	256	1.97
Width/Height	80	160	2.0
		Average	2.0

Table 4.7 represented the actual and pixel values for the 10 possible contact points for the robotic finger to touch the square object for gripping. From the data interpreted in the table, the average contact points accuracy which are 99.32% and 97.34% represented the horizontal coordinate and vertical coordinate respectively.

Table 4.7: Actual and Pixel Values of the Circle Contact Points

Contact Point	**Angle**	**Actual Value (x, y)**		**Pixel Value (width, height)**		**Accuracy (horizontal, vertical)**	
1	45°	220.00	90.00	221.50	87.00	99.32%	96.67%
2	40°	220.00	96.44	221.50	93.50	99.32%	96.95%
3	35°	220.00	101.99	221.50	99.00	99.32%	97.07%
4	30°	220.00	106.91	221.50	104.00	99.32%	97.28%
5	25°	220.00	111.35	221.50	108.50	99.32%	97.44%
6	20°	220.00	115.44	221.50	112.50	99.32%	97.45%
7	15°	220.00	119.28	221.50	116.50	99.32%	97.67%
8	10°	220.00	122.95	221.50	120.00	99.32%	97.60%
9	5°	220.00	126.50	221.50	123.50	99.32%	97.63%
10	0°	220.00	130.00	221.50	127.00	99.32%	97.69%
					Average	99.32%	97.34%

Figure 4.14 demonstrates all the possible contact points in both actual and pixels value for square object. From both graphs, the patterns are also similar. It indicated the 10 possible contact points that the Link 2 of robotic finger will be touched. All of the possible contact points were located on the last quadrant of the square object.

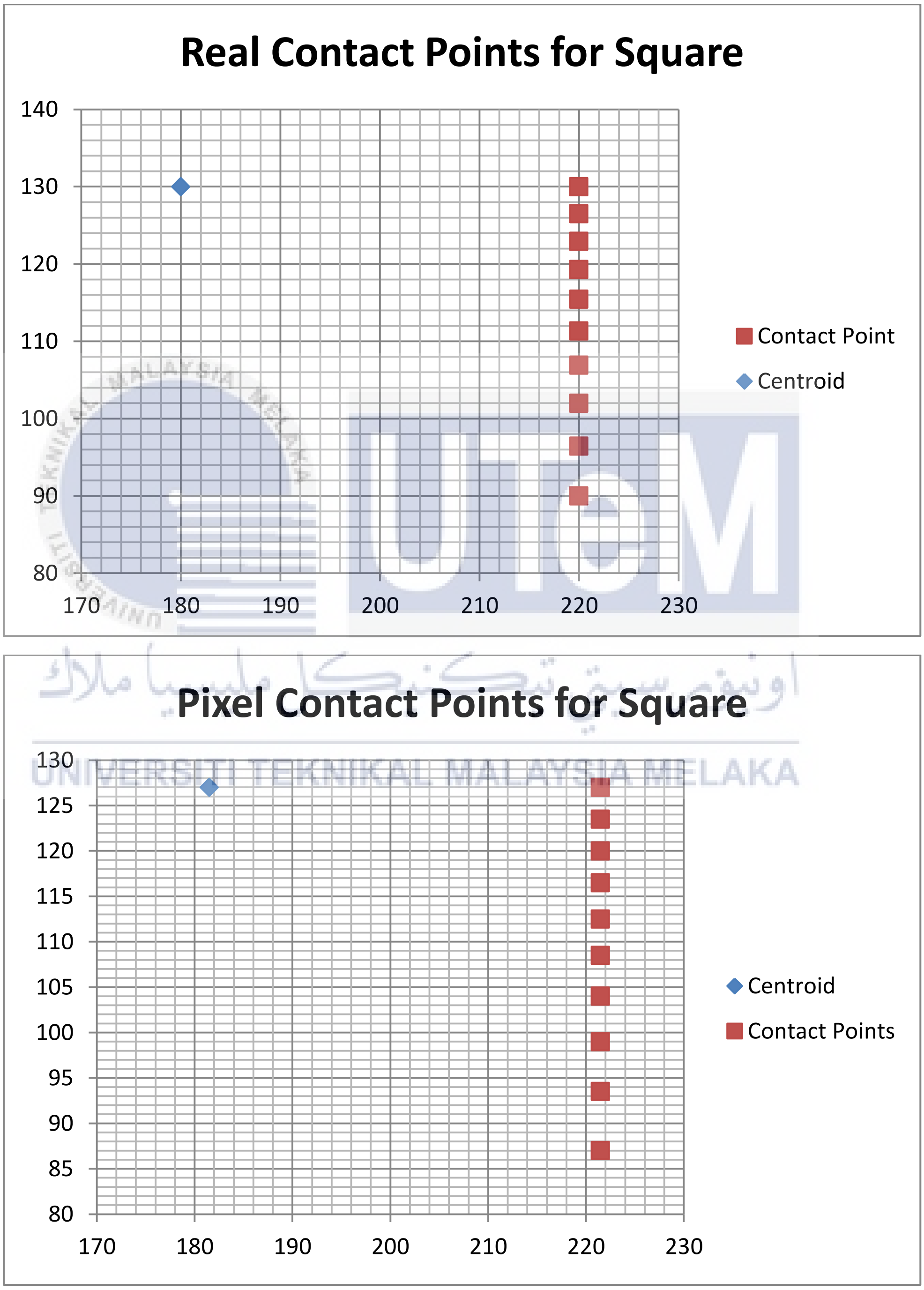


Figure 4.14: All Possible Contact Points in Actual and Pixels Value for Square Object

4.5 Reliability Analysis on Angle Determination

This analysis used all the possible contact points obtained from the previous analysis results. Table 4.8 shows the parameter of actual value and pixel value involved to determine the orientation of Link 1 and Link 2 for grasping both different shapes. The parameters involved are consisting of the origin of the Link 1 as well as the length of Link 1 and Link 2. This parameter mentioned obviously involved in the calculation for finding the orientations of both links to move the robot finger.

Table 4.8: Actual and Pixel Values of the Parameter of Robotic Finger

For Gripping Circle Object			
Parameter	**Actual Value**	**Pixel Value**	**Accuracy**
Link 1	52	52.20	99.62%
Link 2	45	44.88	99.73%
Origin (x-coordinate)	196	191.71	97.81%
Origin (y-coordinate)	97	85.85	88.51%
For Gripping Square Object			
Parameter	**Actual Value**	**Pixel Value**	**Accuracy**
Link 1	52	52	100.00%
Link 2	45	45	100.00%
Origin (x-coordinate)	198	199.5	99.25%
Origin (y-coordinate)	75	72	96.00%

Table 4.9: Parameter Involved in Inverse Kinematic for Circle Contact Points

Contact point	**Angle**	**Actual Value**				**Pixel Value**			
		θ_2	**k_1**	**k_2**	**θ_1**	**θ_2**	**k_1**	**k_2**	**θ_1**
1	45°	60.39°	99.43	47.43	57.46°	63.99°	209.74	102.74	57.79°
2	40°	65.44°	103.40	51.40	54.67°	68.60	217.14	110.14	55.07°
3	35°	70.48°	107.35	55.35	51.99°	72.93	224.10	117.10	52.71°
4	30°	75.49°	111.29	59.29	49.44°	77.83	231.96	124.96	50.16°
5	25°	80.48°	115.21	63.21	47.01°	82.53	239.52	132.52	47.94°
6	20°	85.44°	119.11	67.11	44.68°	87.14	246.92	139.92	46.04°
7	15°	90.40°	123.00	71.00	42.48°	92.27	255.16	148.16	43.91°
8	10°	95.32°	126.86	74.86	40.41°	96.37	261.74	154.74	41.93°
9	5°	100.22°	130.71	78.71	38.46°	101.22	269.53	162.53	40.05°
10	0°	105.11°	134.55	82.55	36.67°	106.11	277.38	170.38	38.50°

Table 4.9 represents the data obtained by the calculation using the inverse kinematic approach for gripping the circle. Table 4.10 compares the result of the orientations obtained for the actual value and pixel value. From the table, the accuracy can be calculated. The accuracy for the orientation of Link 1 is 97.54% while the orientation of the Link 2 is 97.4%.

Table 4.10: Comparison of Orientations for Actual Value and Pixel Value Each Circle Contact Points

Contact point	**Angle**	**Actual θ_1**	**Pixel θ_1**	**Accuracy**	**Actual θ_2**	**Pixel θ_2**	**Accuracy**	**Test**
1	45°	57.46°	57.79°	99.44%	60.39°	63.99°	94.38%	Over
2	40°	54.67°	55.07°	99.26%	65.44°	68.60°	95.41%	Over
3	35°	51.99°	52.71°	98.63%	70.48°	72.93°	96.64%	Over
4	30°	49.44°	50.16°	98.57%	75.49°	77.83°	96.99%	Over
5	25°	47.01°	47.94°	98.06%	80.48°	82.53°	97.52%	Miss
6	20°	44.68°	46.04°	97.05%	85.44°	87.14°	98.06%	Miss
7	15°	42.48°	43.91°	96.75%	90.40°	92.27°	97.97%	Miss
8	10°	40.41°	41.93°	96.37%	95.32°	96.37°	98.91%	Miss
9	5°	38.46°	40.05°	96.04%	100.22°	101.22°	99.01%	Grip
10	0°	36.67°	38.50°	95.25%	105.11°	106.11°	99.06%	Grip
		Average		97.54%	Average		97.40%	

The tests run by implementing all the orientations obtained for each possible contact point measured to touch by the robotic finger. The result acquired from the test stated that the orientations of Contact Point 1, Contact Point 2, Contact Point 3 and Contact Point 4 lead to over-grip the object. The orientations of Contact Point 5, Contact Point 6, Contact Point 7 and also Contact Point 8 are lead to miss gripped the object. Meanwhile, the orientations of Contact Point 9 and Contact Point 10 show the succeeding in gripped the circle object.

Table 4.11: : Parameter Involved in Inverse Kinematic for Square Contact Points

Contact point	**Angle**	**Actual Value**				**Pixel Value**			
		θ_2	**k_1**	**k_2**	**θ_1**	**θ_2**	**k_1**	**k_2**	**θ_1**
1	45°	149.20°	169.18	117.18	-0.42°	107.15°	272.31	168.31	2.57°
2	40°	143.98°	165.08	113.08	9.85°	149.20°	338.37	234.37	9.63°
3	35°	138.71°	160.94	108.94	16.72°	143.93°	330.08	226.08	16.42°
4	30°	133.59°	156.92	104.92	21.65°	138.70°	321.87	217.87	21.40°
5	25°	128.66°	153.05	101.05	25.38°	133.49°	313.69	209.69	25.16°
6	20°	123.89°	149.30	97.30	28.36°	128.49°	305.83	201.83	28.07°
7	15°	119.22°	145.63	93.63	30.84°	123.82°	298.49	194.49	30.61°
8	10°	114.57°	141.98	89.98	32.99°	118.94°	290.83	186.83	32.66°
9	5°	109.90°	138.31	86.31	34.90°	114.50°	283.86	179.86	34.51°
10	0°	105.11°	134.55	82.55	36.67°	109.90°	276.63	172.63	36.23°

Table 4.11 represents the data gained by the mathematical operation by implementing the inverse kinematic for gripping the square. Table 4.12 equates the result of the orientations obtained between the actual value and pixel value. From the table, the accuracy can be calculated. The accuracy for the orientation of Link 1 is 97.48% while the orientation of the Link 2 is 93.8%.

Table 4.12: Comparison of Orientations for Actual Value and Pixel Value Each Square Contact Points

Contact point	Angle	Actual θ_1	Pixel θ_1	Accuracy	Actual θ_2	Pixel θ_2	Accuracy	Test
1	45°	-0.42°	2.57°	85.95%	149.20°	107.15°	71.82%	Grip
2	40°	9.85°	9.63°	97.80%	143.98°	149.20°	96.50%	Fail
3	35°	16.72°	16.42°	98.18%	138.71°	143.93°	96.38%	Fail
4	30°	21.65°	21.40°	98.84%	133.59°	138.70°	96.31%	Fail
5	25°	25.38°	25.16°	99.12%	128.66°	133.49°	96.38%	Fail
6	20°	28.36°	28.07°	98.96%	123.89°	128.49°	96.42%	Fail
7	15°	30.84°	30.61°	99.23%	119.22°	123.82°	96.28%	Fail
8	10°	32.99°	32.66°	99.00%	114.57°	118.94°	96.32%	Fail
9	5°	34.90°	34.51°	98.87%	109.90°	114.50°	95.98%	Fail
10	0°	36.67°	36.23°	98.81%	105.11°	109.90°	95.64%	Fail
		Average		97.48%	Average		93.80%	

The tests were conducted by applying all the orientations acquired for each potential contact point need the touch by robotic finger to grip the object. The result attained from the test stated that all of the orientations of contact point were failed to grip except for the Contact Point 1 which shows the succeeding in gripped the square object.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

First and foremost, a research to develop a gripper using robotic fingers that can recognize different shapes of the object has been done. This analysis covered two major fields in engineering which is robotic and machine vision. The grasping mechanism implemented inverse kinematic approach while the vision system applied to recognize the objects with different shapes. This research achieved all of the objectives by conducting several analyses.

The first objective was to develop the gripper with two robotic fingers. As result, each of the robotic fingers built with two links and two servomotors. The design applied 52 mm and 42 mm lengths for the first link and the second link. So that, the robotic fingers is the gripper with two degree-of-freedom.

As the second objective achieved, the effectiveness of the vision system to successfully detect the shape were depended on the light intensity applied and also the clear acquired image of the object. The inappropriate light intensity may lead to the existence of shadow. So that, it will be disturbed the process on identifying the shape. The size of the object also plays a big role in detecting the shape. The bigger size of object, the clearer acquired image will be obtained.

Last but not least, the priority in this study determined by several analyses conducted which were comparisons that have been made to find the relationship between the real coordination with the pixel coordination. Instead of that, the analysis conducted to determine the accuracy of all the orientations to move the links of the robotic fingers for grasping purpose. The accuracy of the orientations calculated in actual value and pixel value were 97.54% (Link 1) and 97.4% (Link 2) for grasping circle object while 97.48% (Link 1) and 93.8% (Link 2) for grasping the square object.

5.2 Recommendations

For future improvements, the researcher may further the analysis for grasp the object on random location with fixed location of the gripper. In advance, researcher may apply the servo-visual or stereovision approach to conduct the analysis involved in depth which is on three dimensional coordinate. It also can improvise by implementing more degree-of-freedom or amount of fingers used in grasping. Plus, conduct the analysis with other shapes such as triangle, oval, pentagon and etc. Last but not least, the researcher can implement other image processing techniques that collaborate with the artificial intelligence field to recognize the shape in advance.

REFERENCES

[1] International Federation of Robotics, "How Robots Conquer Industry Worldwide," in *IFR forecast: 1.7 million new robots to transform the world´s factories by 2020*, 2017, no. September, p. 25.

[2] A. Shikany, "Machine Vision Trends : Digital Digital," 2016.

[3] D. Eizicovits and S. Berman, "Automatic Graspability Map Generation Based On Shape-Primitives for Unknown and Familiar Objects," *Rob. Auton. Syst.*, vol. 110, pp. 1–11, 2018.

[4] R. Kumar, S. Lai, S. Kumar, and P. Chand, "Object detection and recognition for a pick and place Robot," *Asia-Pacific World Congr. Comput. Sci. Eng. APWC CSE 2014*, 2014.

[5] M. F. Reis, A. C. Leite, P. J. From, L. Hsu, and F. Lizarralde, "Visual Servoing for Object Manipulation with a Multifingered Robot Hand," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 1–6, 2015.

[6] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Preshaped visual grasp of unknown objects with a multi-fingered hand," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 5894–5899, 2010.

[7] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Comput. Electron. Agric.*, vol. 127, pp. 311–323, 2016.

[8] P. Aivaliotis, A. Zampetis, G. Michalos, and S. Makris, "A Machine Learning Approach for Visual Recognition of Complex Parts in Robotic Manipulation," *Procedia Manuf.*, vol. 11, no. June, pp. 423–430, 2017.

[9] V. Lippiello, B. Siciliano, and L. Villani, "Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 73–86, 2007.

[10] Y. Hu, Z. Li, G. Li, P. Yuan, C. Yang, and R. Song, "Development of Sensory-Motor Fusion-Based Manipulation and Grasping Control for a Robotic Hand-Eye System," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1169–1180, 2017.

[11] F. Chao *et al.*, "Enhanced robotic hand-eye coordination inspired from human-like behavioral patterns," *IEEE Trans. Cogn. Dev. Syst.*, vol. 10, no. 2, pp.

384–396, 2018.

[12] F. Sun, C. Liu, W. Huang, and J. Zhang, “Object Classification and Grasp Planning Using Visual and Tactile Sensing,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 7, pp. 969–979, 2016.

[13] A. Vásquez and V. Perdereau, “Proprioceptive Shape Signatures for Object Manipulation and Recognition Purposes In A Robotic Hand,” *Rob. Auton. Syst.*, vol. 98, pp. 135–146, 2017.

[14] D. Guo, F. Sun, B. Fang, C. Yang, and N. Xi, “Grasping Using Visual and Tactile Sensing,” *Inf. Sci. (Ny).*, vol. 417, pp. 274–286, 2017.

[15] L. Chen, P. Huang, and Z. Zhao, “Refining object proposals using structured edge and superpixel contrast in robotic grasping,” *Rob. Auton. Syst.*, vol. 100, pp. 194–205, 2018.

[16] C. Liu, H. Qiao, J. Su, and P. Zhang, “Vision-based 3-D grasping of 3-D objects with a simple 2-D gripper,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 5, pp. 605–620, 2014.

[17] F. Farahmand, M. Pourazad, and Z. Moussavi, “An intelligent assistive robotic manipulator.,” *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 5, no. June 2014, pp. 5028–31, 2005.

[18] M. E. Handbook and S. Edition, “Machine vision 46.,” .

[19] E. R. DAVIES and J. Billingsley, “Machine Vision,” *Mach. Vis.*, pp. xxi–xxii, 2005.

[20] P. J. Sanz, A. Requena, J. M. Inesta, and A. P. Del Pobil, “Grasping the not-so-obvious,” *IEEE Robotics & Automation Magazine,* vol. 12, no. 3, pp. 44–52, 2005.

APPENDICES

APPENDIX A OPENCV-PYTHON SOURCE CODE

take.py

```
import cv2

camera = cv2.VideoCapture(0)
i = 0
while i < 10:
    raw_input('Press Enter to capture')
    return_value, image = camera.read()
    cv2.imwrite('opencv'+str(i)+'.png', image)
    i += 1
del(camera)
```

remove.py

```
import cv2
import numpy as np

## (1) Read
img = cv2.imread("opencv0.png")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

## (2) Threshold
th, threshed = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY_INV|cv2.THRESH_OTSU)

## (3) Find the min-area contour
cnt = cv2.findContours(threshed, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)[-2]
cnts = sorted(cnt, key=cv2.contourArea)
for cnt in cnts:
    if cv2.contourArea(cnt) > 100:
        break

## (4) Create mask and do bitwise-op
mask = np.zeros(img.shape[:2],np.uint8)
cv2.drawContours(mask, [cnt],-1, 255, -1)
dst = cv2.bitwise_and(img, img, mask=mask)

## Save it
cv2.imwrite("pic1.jpg", dst)
```

detect_shape.py

```
from pyimagesearch.shapedetector import ShapeDetector
import argparse
import imutils
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
   help="path to the input image")
args = vars(ap.parse_args())

# load the image and resize it to a smaller factor so that
# the shapes can be approximated better
```

```
image = cv2.imread(args["image"])
resized = imutils.resize(image, width=300)
ratio = image.shape[0] / float(resized.shape[0])

# convert the resized image to grayscale, blur it slightly,
# and threshold it
gray = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]

# find contours in the thresholded image and initialize the
# shape detector
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
  cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
sd = ShapeDetector()

# loop over the contours
for c in cnts:
  # compute the center of the contour, then detect the name of the
  # shape using only the contour
  M = cv2.moments(c)
  cX = int((M["m10"] / M["m00"] * ratio))
  cY = int((M["m01"] / M["m00"] * ratio))
  shape = sd.detect(c)
  print cX, cY, ratio

  # multiply the contour (x, y)-coordinates by the resize ratio,
  # then draw the contours and the name of the shape on the image
  c = c.astype("float")
  c *= ratio
  c = c.astype("int")
  cv2.drawContours(image, [c], -1, (0, 255, 0), 2)
  cv2.putText(image, shape, (cX, cY), cv2.FONT_HERSHEY_SIMPLEX,
    0.5, (255, 255, 255), 2)

  # show the output image
  cv2.imshow("Image", image)
  cv2.waitKey(0)
```

shapedetector.py

```
# import the necessary packages
from pyimagesearch.shapedetector import ShapeDetector
import argparse
import imutils
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
  help="path to the input image")
args = vars(ap.parse_args())

# load the image and resize it to a smaller factor so that
# the shapes can be approximated better
image = cv2.imread(args["image"])
resized = imutils.resize(image, width=300)
ratio = image.shape[0] / float(resized.shape[0])

# convert the resized image to grayscale, blur it slightly,
```

```
# and threshold it
gray = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]

# find contours in the thresholded image and initialize the
# shape detector
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
  cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
sd = ShapeDetector()

# loop over the contours
for c in cnts:
  # compute the center of the contour, then detect the name of the
  # shape using only the contour
  M = cv2.moments(c)
  cX = int((M["m10"] / M["m00"] * ratio))
  cY = int((M["m01"] / M["m00"] * ratio))
  shape = sd.detect(c)
  print cX, cY, ratio

  # multiply the contour (x, y)-coordinates by the resize ratio,
  # then draw the contours and the name of the shape on the image
  c = c.astype("float")
  c *= ratio
  c = c.astype("int")
  cv2.drawContours(image, [c], -1, (0, 255, 0), 2)
  cv2.putText(image, shape, (cX, cY), cv2.FONT_HERSHEY_SIMPLEX,
    0.5, (255, 255, 255), 2)

  # show the output image
  cv2.imshow("Image", image)
  cv2.waitKey(0)
```

edge.py

```
import cv2
import numpy as np

img = cv2.imread('circle3.jpg',0)
edges = cv2.Canny(img,60,255)

indices = np.where(edges != [0])
coordinates = zip(indices[0], indices[1])

print coordinates

f=open('aqmal.txt', 'w')
f.write(str(indices[0]))
f.write(' ')
f.write(str(indices[1]))
f.write('\n')
f.close()
```