THE DEVELOPMENT OF OBJECT MEASURING SYSTEM USING IMAGE PROCESSING TECHNIQUE

This report is submitted in accordance with requirement of the University Teknikal Malaysia Melaka (UTeM) for Bachelor Degree of Manufacturing Engineering

by

SEAH MEI YIN

B051410154

940926-04-5166

FACULTY OF MANUFACTURING ENGINEERING

2018

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN PROJEK SARJANA MUDA

Tajuk: **THE DEVELOPMENT OF OBJECT MEASURING SYSTEM USING IMAGE PROCESSING TECHNIQUE**

Sesi Pengajian: **2017/2018 Semester 2**

Saya **SEAH MEI YIN (940926-04-5166)**

mengaku membenarkan Laporan Projek Sarjana Muda (PSM) ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka (UTeM) dengan syarat-syarat kegunaan seperti berikut:

1. Laporan PSM adalah hak milik Universiti Teknikal Malaysia Melaka dan penulis.
2. Perpustakaan Universiti Teknikal Malaysia Melaka dibenarkan membuat salinan untuk tujuan pengajian sahaja dengan izin penulis.
3. Perpustakaan dibenarkan membuat salinan laporan PSM ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. *Sila tandakan (√)

☐ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysiasebagaimana yang termaktub dalam AKTA RAHSIA RASMI 1972)

☒ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/ badan di mana penyelidikan dijalankan)

☐ TIDAK TERHAD

Disahkan oleh:

Alamat Tetap: 60, JALAN DUKUL 11, TAMAN RUMPUN BAHAGIA, 75300 MELAKA	Cop Rasmi:
Tarikh: 2nd JUNE 2018	Tarikh: ________________

*Jika Laporan PSM ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan PSM ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I hereby, declared this report entitled "The Development of Object Measuring System Using Image Processing Technique" is the result of my own research except as cited in references.

Signature : ……………………………………
Author's Name : SEAH MEI YIN
Date : 2nd June 2018

APPROVAL

This report is submitted to the Faculty of Manufacturing Engineering of Universiti Teknikal Malaysia Melaka as a partial fulfilment of the requirement for Degree of Manufacturing Engineering (Engineering Materials) (Hons). The members of the supervisory committee are as follow:

………………………………………………

(Dr. Ruzaidi Bin Zamri)

……………………………………

(Dr. Mohd Sukor Bin Salleh)

ABSTRAK

Sistem ukuran secara manual telah digunakan dalam bidang pertanian, pakaian, perubatan, antropometrik dan pembinaan. Sistem ini memerlukan ketepatan yang sangat tinggi dalam unit tertentu untuk mengelakkan masalah produk tidak boleh muat. Walau bagaimanapun, system ukuran secara manual tidak berfungsi seperti yang disangka kerana cara ukuran ini memerlukan masa yang banyak, keputusannya tidak tepat dan orang yang melakukan ukuran perlukan penglihatan yang tajam. Dalam automasi era, semua progress termasuk sistem ukuran dijangka tamat dalam masa yang lebih singkat dan memperoleh ukuran ketepatan yang tinggi. Oleh sedemikian, banyak penyelidik melakukan pengajian berkaitan dengan system ukuran. Pengukuran manual semakin diganti dengan kaedah pengukuran baru yang dikenali sebagai pengukuran penglihatan. Dalam era automasi sekarang, data antropometrik juga amat penting dalam gaya hidup harian manusia disebabkan oleh perluasan penggunaannya dari penggunaan tentera, industri pakaian kepada peranti elektronik seperti telefon pintar. Oleh itu, matlamat utama projek ini adalah untuk membangunkan sistem pengukuran penglihatan untuk mengukur lingkar tangan. Gabungan sistem perkakasan dan perisian dapat mengotomatisi kaedah ukuran lingkar tangan untuk menggantikan pengukuran manual. Sistem perkakasan termasuk kamera, papan induk dan komputer riba manakala sistem perisian merujuk kepada sistem pengaturcaraan. Biasanya, pengekodan untuk sistem pengukuran penglihatan tidak akan didedahkan. Sehubungan dengan itu, sumber terbuka Python digunakan sebagai bahasa pengaturcaraan untuk projek ini. Kadar berjaya untuk pengukuran penglihatan mencapai 80% dan pengukuran secara manual mencapai 100%. Perbandingan kaedah ini dalam penggunaan masa telah membuktikan pengukuran penglihatan lebih cepat. Analisis dan perbandingan antara kaedah ini mengesahkan bahawa pengukuran penglihatan boleh menggantikan pngukuran manual kerana hanya subjek number lapan bermasalah. Projek ini dijangka memperkenalkan cara automatik menggunakan pengukuran penglihatan untuk menggantikan pengukuran manual dalam usaha mendapatkan data antropometrik.

ABSTRACT

Measuring system has been used in agriculture, apparel, medicine, anthropometry and construction since the born of the manual measurement. It requires high quality of accuracy to reduce the error of unfit in the finish product. However, manual measurement is not very practical because it has been proved to be time consuming, lack of accuracy and it requires good eyesight. Due to the drive of the high demand in accuracy and shorter time in the automation era, a larger and deeper research in improving measuring data has been done. Manual measurement slowly replaced by a new method of measuring system, which is known as vision measurement. In this current automation era, anthropometric data also becomes critical in the daily lifestyle of human beings. This is because its usage has been widened from the military used, apparel industry to electronic device such as smartphone. Therefore, the main objective of this project is to develop a vision measurement system to measure hand palm. A combination of hardware and software system will fully automate in getting the measurement data to replace manual measurement. Hardware system includes a camera, a mother board and a laptop while software system refers to the programming developing system. Normally, the coding for the well-developed vision measurement system will not reveal. Hence, open source Python is used as the programming language for this project to demonstrate the automation in measurement. Finally, data collection for both methods is done and the successful rate for vison and manual are 80% and 100% respectively. Comparison in terms of time consumption proved that vison method is faster than manual method. Lastly, analyzed and compared these two type of measuring system concluded that vision method can replace manual method because only data subject number eight has error. An improvement and further analysis in future can minimise and eliminate this problem. Therefore, this project is expected to introduce an automatic way using vision measurement to replace manual measurement to get anthropometric data.

DEDICATION

Special thanks to my one and the only

my beloved parents, Seah Tiong Sui and Lim Beng Huat

and to my adored sisters,

for giving me moral support, money, cooperation, encouragement and also understandings.

Thank you to all my friends who willing to support me in data collection.

I also want to say thank you to my kind and helpful supervisor, Dr Ruzaidi Bin Zamri

for your guidance and support at all times.

Lastly, appreciation goes to my knowledgable and experience examiners

for spending your precious time reading my draft and corrected me during my presentation.

ACKNOWLEDGEMENT

Thank you for my surrounding people and my strong determination and spirit I manage to complete this final year project successfully by solving all the difficulties one by one. It was really a long process with challenging hardship.

My respected supervisor, Dr. Ruzaidi Bin Zamri. His kindness, unwavering patience and mentorship guided me through the process, his easily understood explanations and open mind allowed me to grow and learn in such a way that I am now a better researcher. Besides that, I would like to express my gratitude to the library staff and all the lecturers for their kind support in different way, advice and guidance as well as push me back to the correct direction when I was lost throughout the study.

Also, I would like to give a special thanks to my best friends who gave me much motivation and cooperation mentally in completing this report especially when I really wanted to give up and change title due to the stress of unable to solve the difficulties. They had given their critical suggestion and comments throughout my research. Thanks for the great friendship.

Finally, I would like to thank everybody who play an important role in supporting me to complete this FYP report.

TABLE OF CONTENT

CHAPTER 5: CONCLUSION AND RECOMMENDATION

APPENDIX

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BC	-	Before christ
SI	-	Système internationale / International system
PSFS	-	Patient-specific functional scale
GUI	-	Graphic user interface
3D	-	3-Dimensional
2D	-	2-Dimensional
ROI	-	Region of interest
RGB	-	Red green blue
BGR	-	Blue green red
TI	-	Thermal index
MRI	-	Magnetic resonance imaging
CT	-	Computer tomography
HSL	-	Hue-saturation-luminance
EMT	-	Edge maximum technique
IEEE	-	Institute of electrical and electronics engineers
UTeM	-	University Teknikal Malaysia Melaka
RPi	-	Raspberry Pi
GPU	-	Graphics processing unit
OpenCV	-	Open source computer version
CPU	-	Central processing unit
RAM	-	Random access memory
HDD	-	Hard disk drive
LAN	-	Local area network
FDH	-	Fibre distribution hub
LED	-	Light emitting diode
OS	-	Operating system
SL	-	Single language
WiFi	-	Wireless fidelity
BLE	-	Bluetooth low energy

GPIO	-	General purpose input/output
USB	-	Universal serial bus
HDMI	-	High definition multimedia interface
CSI	-	Camera serial interface
SD	-	Secure digital
FYP I	-	Final year project I
FYP II	-	Final year project II
FFC	-	Flexible flat cable
VNC	-	Virtual network computing
HPF	-	High-pass filters
LPF	-	Low-pass filters
VS	-	Versus
Jpg	-	Joint photographic expert group
STD	-	Standard deviation
LOA	-	Limit of agreement
LED	-	Light emitting diode
BOM		Bill of material
ECM		Environmental conscious manufacturing
LLL		Long life learning
BE		Basic Entrepreneurship

LIST OF SYMBOLS

BC	-	Before Christ
SI	-	Système Internationale / International System
%	-	Percentage
π	-	Pi
r	-	Radius
mm	-	Millimetre
$\sum$	-	Summation
$\bar{x}$	-	Mean

CHAPTER 1

INTRODUCTION

1.1 RESEARCH BACKGROUND

Measurement system provides unit to outline the identity and characteristic of a product. The identity and characteristic include value, size and shape of the product. It is widely used and great contributes in agriculture, education, construction, food and clothes industry. Measurement system was originated in the 3rd millennium BC. This was the period when human beings started to transform from a moving to a nomadic lifestyle. With a replacement of money to trade from exchange system, known as "Barter System", the measurement system became critical. The slowly growing of trading importance between community and across the world elevated the growing of measurement system to a higher accuracy and increasingly diverse set of fields. The first of measurement technique and standard were created during the Egyptians time when Royal Cubit (Figure 1.1) was use as today's length measurement. It was a technique measure from the distance from Pharaoh's elbow to his fingertips (David Flack, October 2012). Today, the science of measurement term is known as metrology. It involves dimensional measurement which defines as a measuring of an artefact. Artefact can be the diameter of a ball, the length of a table and the volume of a bottle. In metric world, the SI unit of length is meter and it is possible to switch to millimeter, centimeter, kilometer and other related measurement (Flack, 2009).

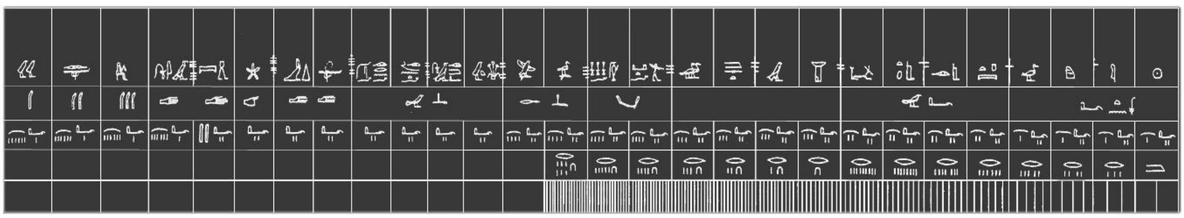


Figure 1.1: Royal Cubit

To study a variety of human’s aspects within past and present societies, an anthropologist engages in the practice of anthropology. Human body measurement is collected to complete the anthropometric data of community. The data collected is useful for military database, educational and future research. The future research includes a previous study to investigate the reliability of Patient-Specific Functional Scale (PSFS) by measuring the PSFS with a standard error to justify the reliability of the clinical utility (Wright et al., 2018). There are also plenty of ergonomics researcher seeks to improve the ergonomics data to study deeper on the behavior of specific community. A group of researchers came up with an idea using the hand dimensions and index finger length ratio to determine the gender. They concluded using hand dimensions can detect the gender in forensic identify investigation (Ibrahim et al., 2016). In 2016, researchers measured the hand morphometry at different gestational ages babies. They said the data is important for industrial applications especially in gripping performances of pre-term baby things (Honoré et al., 2016).

With the rapid growing of technology in modern era, the manual measuring instrument slowly replaced by digital measuring instrument and now vision measuring instrument. Two-dimensional (2D) photogrammetry was served as the primary sources for craniofacial measurement data to covered the inaccuracy of direct anthropometry and it was then transformed to three-dimensional (3D) imaging technique such as laser surface scanner and stereo-photogrammetry for craniofacial investigation (Weinberg et al., 2004). A hand anthropometry survey was carried out and electronics digital caliper with an accuracy of 0.01mm was used to measure twenty-four hand dimensions (Mandahawi et al., 2008). Vision sensing and image processing compile with machine visual identification technology is largely used in automation industry due to its fast speed and high accuracy (Min & Principle, 2015). It must comply with the application of digital image processing and a computer to transfer an image or video from the camera and then using algorithms to measure the subject parameter or track the motion of an object in the image. Using a combine method of photogrammetry and digital image processing can make real time traffic measurement to adjust the timings of traffic lights (Zhu et al., 2015). Jing Min mentioned that a new detection technology using machine vision can replace traditional techniques of detecting screw thread (Min & Principle, 2015).

Vision measuring method is the current most effective advance method and most of its application use in the field of automation industry to speed up the parameter measuring of components of electronics, appliances or material. It can be very useful especially in anthropometric and clinical categories. For example, digital image processing technique measured the torsion from eye multitemporal based on the eye image (Parker et al., 1985).

1.2 PROBLEM STATEMENT

The method of measuring 3 selective parts of hand, which are palm breadth, palm length and finger length is still using Vernier caliper (Figure1.2) or digital caliper (Figure 1.3). It is not surprised that every single measurement is subject to uncertainly (Paolo et al., 2017). There are many type of errors occurred proven by previous research stated that direct measurement is not reliable. According to the authors of Fundamentals of Dimensional Metrology book, caliper is difficult to follow Abbe’s Law to achieve high accuracy because there are always errors (Flack, 2009). Errors appears with the reason of the untreated experimental measuring apparatus intervals (Krechmer, 2018) In measurement, risk of human error is associated with operator error which includes physical and mental stress state that lead to observational error and wrong formal procedures as well as response options (Paolo et al., 2017). Reading the calliper requires good eyesight and skill to avoid misreading (Flack, 2009). The manufacturers can only manipulate the error to minimum but they cannot eliminate the error (Dotson, 2016).

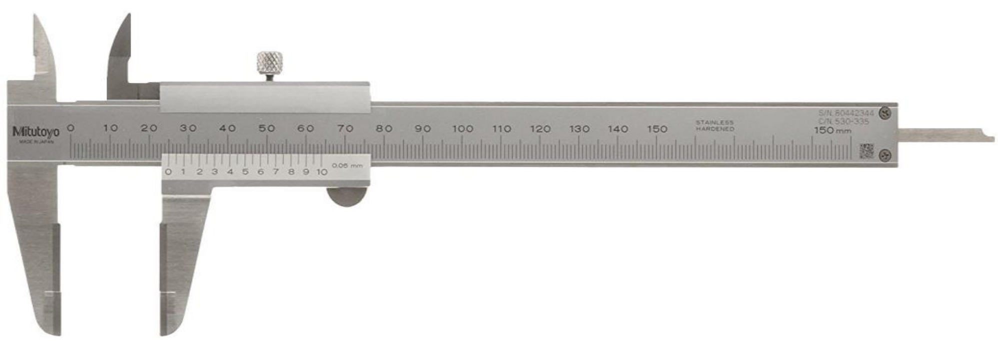


Figure 1.2: Vernier caliper

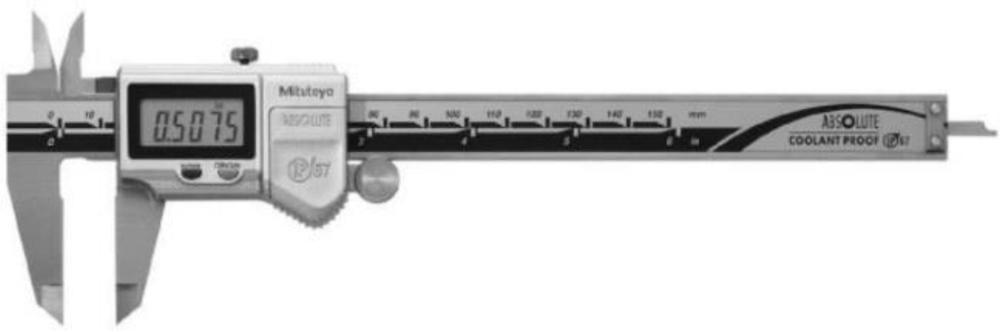


Figure 1.3: Digital caliper

Besides, proceeding a measurement using caliper is very time consuming (Kohnen, 2002). The lack of accuracy and time consuming may become main issue in automation era. Hence, there is a need to replace this traditional manually measuring hand method with a faster, higher accuracy and easier method. Not only that, many researchers do not reveal their measurement coding. Therefore, that is a need to demonstrate automation in measurement. Coding is developed with the used of open source software known as Python.

1.3 OBJECTIVES

The objectives are as follows:

(a) To develop an image processing coding for hand breadth, hand palm and finger length measurement and compute finger vs palm (shape) index.

(b) To validate vision sensor measurement method with manual measurement method in term of time consumption.

(c) To compare vision sensor measurement method with manual measurement method.

1.4 SCOPES OF THE RESEARCH

The scopes of research are very important and there is a total of three scopes. First, this research is using Python software to develop image processing coding. Python is a well-known open source software can decrease the cost spending.

Second, research on vision measurement focuses only on hand breadth and palm length as well as finger length. The measurement data will be tabulated. Finger vs. palm (shape) index will be calculated using data of finger length and palm length and palm breadth for every sample. This research on human body measurement can contribute towards the ergonomics and anthropometric especially in collecting human body measurement data. Figure 1.4 shows the finger length, palm breadth and palm length.

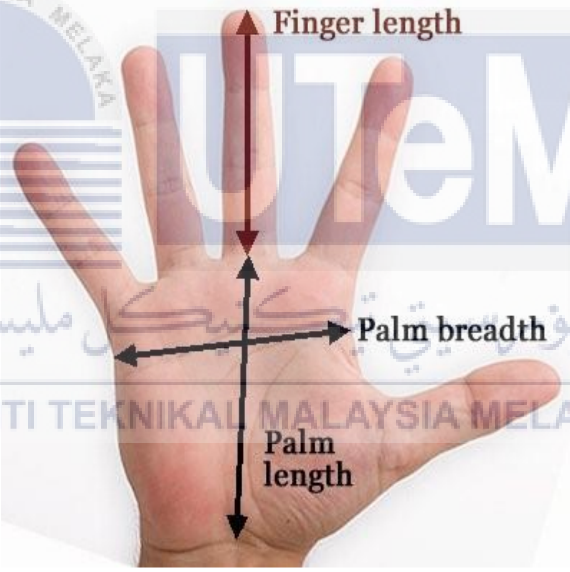


Figure 1.4: Finger length, palm breadth and palm length.

Third, all the data collected only focus on Asia people. The Asian sampling collected is exactly thirty and the subjects are randomly selected. Besides, the data only based on the right hand to standardise the data collected. The raw data collected can be used for analysis including comparison in method collected between machine vision and manually measurement.

1.5 SUMMARY

The entire research consists of 2 main parts. The first part can be classified as research background studying and it consists of chapter 1, 2 and 3. Second part is output and summary and this part only covers in chapter 4 and 5.

The first part emphasizes on studying and digging previous relevant information or data to outsource a general idea on how the project can be proceed provided with strong reference and structural work flow. There is no limit for the relevant information. The information can be found in the form of journal, book, magazine and newspaper with a condition it is qualified in the standard. After gathering of data, a basic contents extraction and tabulation needed to support this thesis. Purpose of carried out this thesis can be justified with a strong solving method and the scope of thesis can be figured out. Chapter 1, Introduction is introducing the background, problem statement, objective and scope. Chapter 2 is Literature Review. This chapter describes the previous research to solve the related problem and output a general idea how to solve the problem of this thesis. The table of listed journal will be included in Appendix A and Appendix B. Chapter 3 is Methodology which illustrates the procedure of thesis in the form of work flow and Gantt chart. Gantt chart can be found in Appendix C and Appendix D.

Second part shows the output using digital image processing technique and the traditional manually measurement technique to obtain the raw Asian sample for further data comparison and analysis to mark the differences of both techniques and to ensure the digital image processing is stable and reliable to function as expected. Then, writing a conclusion to summarize the result obtained. An approximately 30 sample data of hand breadth and palm are collected. Chapter 4, Result and Discussion explains the result obtained with description, graphs and pictures are attached. A complete source code for the programming can refer to Appendix E. Chapter 5, Conclusion and Recommendation highlights the output of the thesis and conclude this thesis with some valuable recommendation to support future research.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter focuses only the theory and research from the other researchers. All the journals show on this chapter have been defined and approved years ago. Important and relevant information from previous studies are extracted as guideline and reference. Basically, all the collected information will be categorized into five categories. The first describes the manual measurement and it consists of measuring instruments and issues of manual measuring as sub-categories. The second describes the vision measurement. It is followed by digital image processing technique which will be divided into another three sub-categories, image (grayscale) segmentation, color segmentation as well as edge and line detected segmentation. Then, the last part is an overview of past researchers from journals and reference papers which are listed in tabulated form.

2.2 MANUAL MEASUREMENT

Manual Measurement means direct measure using human power with a suitable measuring tool. It is a traditional way for anthropometric measurement such as palm length, palm breadth and finger length. A study shows that hand breadth, palm length and finger length were the three main factors cause the 78.3% of variance in hand shape variability (Jee & Yun, 2016). Anthropometry and agricultural hand tool are strongly related and they proof that good fit in hand was one of the top preferable tool characteristics (Wibowo & Soni, 2014). A study describes that direct measurement tools were traditionally used to obtain the hand dimensions for glove industry (Garrett, 1971).

2.2.1 Measuring instruments

Jee & Yun (2016) carried out the measurement for length, breadth and thickness of hands and fingers using digital calipers and for the hands circumference and finger joints, they used tape measuring method. Martin anthropometer is used for direct measurement in a study (Park et al., n.d.). Figure 2.1 shows a martin anthropometer.

Figure 2.1: Martin anthropometer (Park et al., n.d.)

Mandahawi et al. (2008) used an electronic digital caliper with an accuracy of 0.01mm to measure twenty-four hand dimensions in both experiments. To measure the hand data for human babies at birth, from 26 to 41 weeks estimated gestational age, a medical caliper is used and a benchmark measures is important in hand morphometry especially industrial applications (Honoré et al., 2016). Hence, it is obvious that traditional hand measuring method usually only involve caliper and measuring tab for length measurement.

2.2.2 Issues of manual measuring

Using measuring tools to measure can result in low accuracy because the complex anatomy and curvature of thumbs, fingers and related joints (Yu et al., 2013). Another study highlighted the accuracy is mainly relied on the person who did the measurement (Park et al., n.d.). Simmons and Istook (2003) describe the most troublesome source of anthropometric error is the observer error which include landmark position, subject location and instrument applications. This reason is further supported by a study mentioned the accuracy of manual measure is said to be not accurate because it will not only be interfaced by the measurement skills, experience and position mark of the person in charge but also the breathing and posture of the testers (Qi et al., 2011). Besides, a study from Xi'an Polytechnic University proved that error of manual measurement exists and it is caused by the subject or the person in charge. For tester includes elasticity of human body, standing posture, tense of body, breathing and for person in charge to measure will include tools managing calibration, measurement skills experience and position mark (Qi et al., 2011). Physical and mental stress state of an measuring operator leads to observational error and wrong formal procedure as well as response option (Paolo et al., 2017). It is also not surprised to find out that many researchers never had a formal education in anatomy to identify a certain landmarks and this leads to incorrect landmarks for measurement on human body (Simmons & Istook, 2003).

A simple manual anthropometric method is time consuming and not accurate (Li et al., 2013). The same issue was found in 3D Hand Anthropometry of Korean Teenager's and Comparison with Manual Method and the authors further described the manual measuring method is very time consuming (Park et al., n.d.). In 1988, an anthropometric survey of Army personnel used four hours to physically landmark, measure and record data of a subject (Greiner, 1991).

Having initiative trying to improve the accuracy, Mohammad (2005) described he had to use the sliding caliper and steel tape cautiously to take the readings to improve the accuracy. Taking several readings for the same measurement and the average reading will be used as final measurement result also is used in a study mentioned that inaccurate hand dimensions causes re-measurement and repeated adjustments must be taken in therapy treatment in hospital (Yu et al., 2013). Ibrahim et al. (2016) repeated the hand measurement using an anthropometer and a sliding caliper twice again if the initial two readings failed to satisfy 0.4 range criteria besides the measurements should obtain in a well-lit room to eliminate environmental factor.

However, repeated measuring creates more problems. Glove making progress is very slow and person in charge often include 'trial-and-error' and his experience (Yu et al., 2013). A study shows that anthropometric measurement using traditional method is not reliable due to the variability of repeated measurement of the same subject between difference observers

(Stomfai et al., 2011). Mandahawi et al. (2008) highlighted relatively huge absolute percentage difference can caused by measurement systematic error or systematic difference in measurement techniques in a study regarding the hand anthropometry survey for Jordanian population.

The following disadvantage of manual measurement is majority of people prefer no physical contact for a large number of measuring on human body (Simmons & Istook, 2003). Furthermore, a study concluded using one type of measuring tool alone, the data will be overestimated or underestimated (Agarwal & Sahu, 2010).

2.3 VISION MEASUREMENT

Vision measurement is an indirect method for parameter measurement. It is a non-destructive method to the subject and this operation can be used in agricultural and food processing industries (Hossein Mirzabe et al., 2017). This modern measurement is using vision and machine vision technique with the help of image processing software (Parveen, 2017). Basically, the hardware is used to capture images and a developed software is used to perform measurement (Gadelmawla, 2017). A paper written by Massiris et al. (2015) described to perform vision measurement only requires a laptop or portable computer, a camera and designed software in the computer to perform the measurement. Acquisition of image must locate in a controlled environment to avoid dust and stray light (Momin et al., 2017). After the image of target is captured, an image processing algorithm is used to collect data and validate the purpose of research (Parveen, 2017). Since the whole system is an integrated system with a combination of optical, electronic, mechanical and computer technology, the condition and requirement for every single devices should meet to ensure the communication and flow of the framework (Min & Principle, 2015). It is either using three-dimensional (3D) or two-dimensional (2D) image to analyze the body dimension (Yu et al., 2013).

Currently, vision measurement has developed into a new detection technology (Min & Principle, 2015). It can apply in many fields including agriculture, geometry and inspection assurance, infrastructure and environmental control as well as anthropometric measurement. In agriculture, Parveen et al. (2017) proposed to identify the quality of the rice grain by detecting the length, width, area, aspect ratio, color features and chalky based on optical and image processing technique. A study dedicated the vision measurement is useful to investigate the properties of cucumber seeds and kernels based on the moisture content on gravimetrical and frictional properties (Hossein Mirzabe et al., 2017). Images captured can identify the different phenological phases of vegetation cover according to the color indexes (Kırcı, Güneş, & Çakır, 2014). To automate the grading of mangos based on geometry and shape, a group of researchers proposed the use of image processing to replace the manual grading process (Momin et al., 2017).

For geometry and inspection assurance, Jing Min (2015) mentioned the screw thread parameter can be measured by analyzing the image measurement. Gadelmawla (2016) analyzed the screw thread features to maintain the accuracy of measurement with the standard value using vision system measurement. In spring 2005, Kenkare & May-Plumlee (2005) concluded modified digital method using capturing image and further analyzed it can evaluate drape fabric and it is a viable alternative to the traditional conventional cut and weigh processes. Besides, visual measurement is used in proprioceptive shape signatures through generating the 3D model of the target item and analyzing the shape and geometry to identify the position for the robot hand to grasp (Vásquez & Perdereau, 2017). A paper suggests the combination of application of 3D imaging and cloud-to-mesh algorithm can reduce the disturbances and geometrical error stemming on production system (Berglund et al., 2016). In addition, it was once a big problem to ensure the high temperature semi-finished product or object controls in standard shape and parameter before the introducing of camera and software application for hot objects measurement approached (Zatočilová et al., 2016). An optical pipette measurement to measure the glass micropipette tip with high resolution, accuracy and repeatability is an initial step heading fully-automated micropipette fabrication system (Stockslager et al., 2016).

Having a demand in infrastructure and environmental control, Sarikan et al. (2017) described how to apply image processing and computational in automated vehicle classification in dedicated lanes to control the speed variations in traffic. To ease the lives of handicapped people, a device using image processing allows communication between handicapped people and computer to input the signals as interface device to control the environment such as switching on and off of the electrical devices (Takami et al., 1996). Application of photogrammetry and digital image processing can detect the vehicle headlights to measure the traffic queue length on real-time to adjust the timings of traffic lights to monitor the traffic status (Zhu et al., 2015).

Anthropometric measurement is another essential measurement category because it is needed for human use product and anthropometry data of residents from different regions of the world. Simmons and Istook (2003) highlighted the system of vision anthropometric measurement consists of projectors and cameras in the framework for body-scanning propose. This application can be used in various fashion and medical products (Park et al., n.d.). Good-fitting custom-made gloves for individual can be archived using 2D and 3D anatomical image scanning and analyses the hand anthropometric measurement (Yu et al., 2013). High accuracy of vision measurement also applied to improve the ergonomic design of tools (Massiris et al., 2015). In medical field, vision measurement can be used to measure ocular counter-rolling, which is the torsion of the eyes to indicate the otolith function in order to study the motion sickness, a problem in space travel of the patient (Parker et al., 1985). For apparel industry, customized insole and clothes for individual can be utilized directly based on vision-based human body measurement method without body landmark information (Uhm, Park, & Park, 2015)

Having automatic vision measurement system benefits the society and ease the anthropometric measurement progress. A study mentioned automatic human body measurement system using vision-based method is very convenience to tailors and protects customers' privacy and provides comfort and lower in cost (Uhm et al., 2015). It provides significant improvement in fitting comfort of human wearing product (Park et al., n.d.). Besides, this type of anthropometric measurement requires a smaller space and no physical contact (Uhm et al., 2015). With the use of image processing, some complex and highly difficult part such as the ocular torsion of eye can be measure and analyzed (Parker et al., 1985). Body image reconstruction for body surface measurement contributes in obesity and fitness evaluation (Li et al., 2013).

This 3D anthropometry measurement is an alternative method to replace manual anthropometry measurement (Yu et al., 2013). Hence, it can use to replace the traditional way for anthropometric measurement such as palm length, palm breadth and finger length.

2.4 DIGITAL IMAGE PROCESSING TECHNIQUE

Image processing is used for raw image appearance enhancement and with the additional use of analysis, the function can extent from feature extraction, recognition of image to coding, filtering and restoration (B.Dutta Majumder, June, 2011). The raw images are received from satellite, space probes or normal single daily used camera (Chitradevi & Srimanthi, 2014). A study shows the image based methods can solve the time consuming of video processing approach (Sarikan et al., 2017). Basically, measurement using vision requires a digital camera, digital computer and simple set-up tools (Stancic et al., 2009). The digital computer provides display screen and mass storage to process the segmentation algorithm (B.Dutta Majumder, June, 2011).

A digital image on the screen has x and y two-dimension discrete function which representing spatial coordinates, f(x,y) and their magnitude of intensity value or gray level (Relf, 2004). The intensity value can be used to locate a matrix indices which actually refer to a point, known as pixel, image elements or pels but pixel is the most common used (Chitradevi & Srimanthi, 2014). Pixel depth has 8 bit, 16 bit and 32 bit and they are used to show the intensity extremities in a grayscale image (Pal & Pal, 1993). In short, digital image processing is using a digital computer to process a two-dimensional image (Chitradevi & Srimanthi, 2014)

Segmentation is the initial step to carry out object measurement in the image. It is also defined as the primary important step of low level vision (Pal & Pal, 1993). Segmentation technique means divided an image to extract only the "meaningful" region (Zhang, 1996). There are two approaches to present segmentation. The first method is known as grayscale and colour segmentation and the second method is edge detection and segmentation (B.Dutta Majumder, June, 2011). This low level segmentation is considered as the base for the subsequent high level operation such as region merging (Ning et al., 2010).

Basically, there are two types of segmentation algorithms, which are analytical method and empirical method (Zhang, 1996). A segmentation algorithm must be created to do image analysis and it is unique for every image because it cannot perform reasonably well for all images (B.Dutta Majumder, June, 2011). Figure 2.2 shows a general scheme for segmentation and its evaluation.

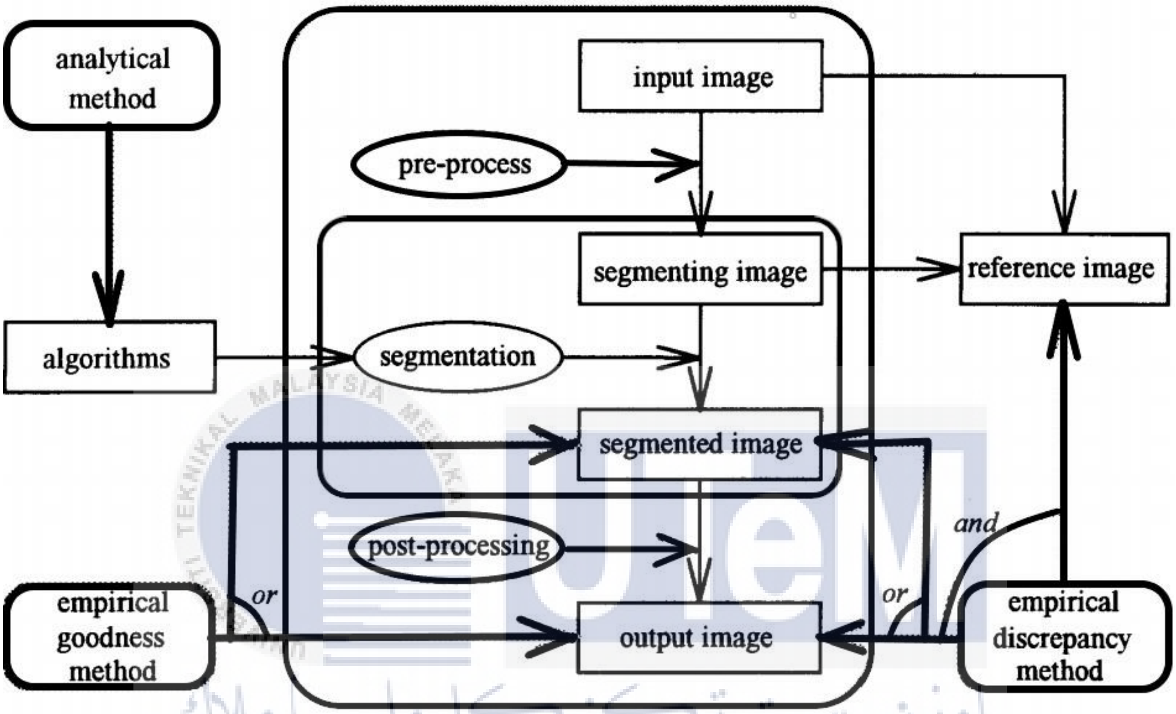


Figure 2.2: General scheme for segmentation and its evaluation (Zhang, 1996)

2.4.1 Image (grayscale) segmentation

Grayscale segmentation approach subdivides the region of interest (ROI) as a complete picture and depicts the homogeneity feature in value (B.Dutta Majumder, June, 2011). Gray segmentation technique is considered the simplest and easiest technique for image extraction (N. Sharma et al., 2010). Gray tone images segementation research and literature around the world is largely more than the color image segmentation (Pal & Pal, 1993). Basically, grayscale is using the concept of a histogram to present the respective light intensity value (pixel level) on z-axis (Relf, 2004). This gray level histogram is known as point dependent techniques and in thresholding (Pal & Pal, 1993). Also, B. Chitradevi and

P.Srimathi (2014) mentioned the gray level for all objects are the same and the gray scale for background is another gray level. This technique is the most suitable technique for an object with uniform brightness against its background in an image to show the gray level (N. Sharma et al., 2010). Different peaks show in the histogram indicate the valleys separated regions in an image (Pal & Pal, 1993). To improve image segmentation, B. Chitradevi and P.Srimathi (2014) said that image enhancement sometimes is needed because the image captured when the imaging sub system and illumination conditions are poor.

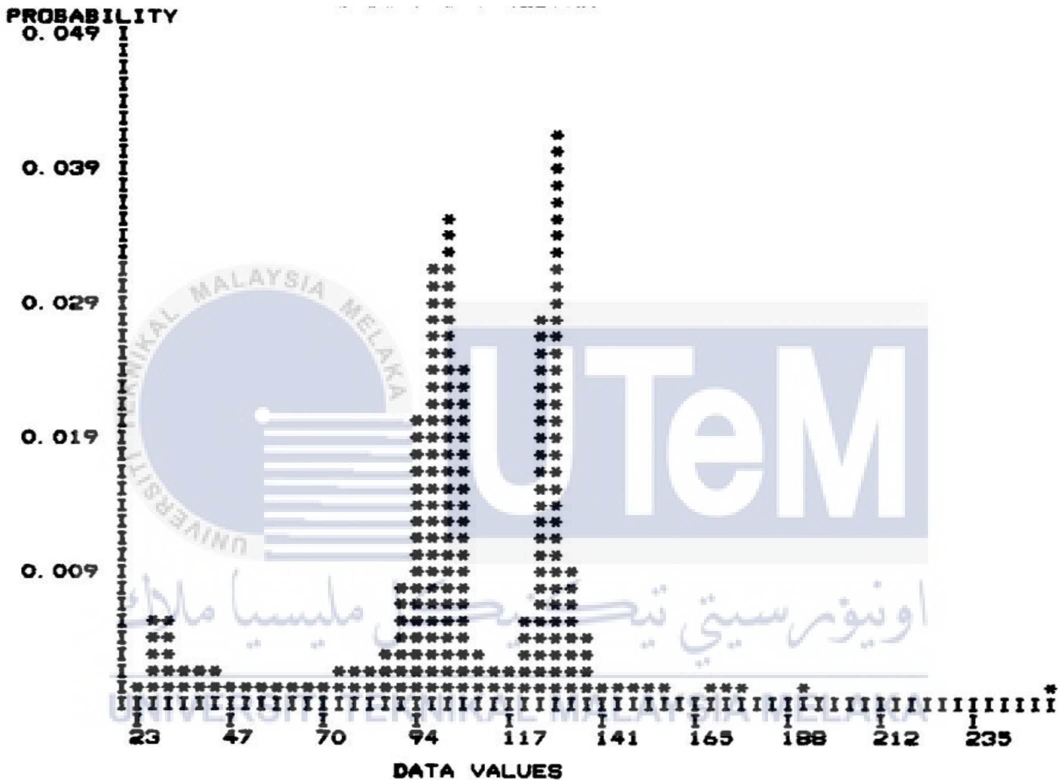


Figure 2.3: Histogram of image (Haralick, 1985)

Figure 2.3 shows an example of histogram of image. The earliest method to use graph for segmentation was to use local a fixed threshold and a local measures (Felzenszwalb & Huttenlocher, 2004). Based on the histogram, the peaks shown are known as maxima and it is usually divided by minima (N. Sharma et al., 2010). Normally, the y-axis is measured as frequency and the x-axis is measured as grayscale in a histogram (B.Dutta Majumder, June, 2011). The histogram is presented using the dissimilarity of the pixels such as intensity, color, motion, location and local attribute linked by edge in image (Felzenszwalb & Huttenlocher, 2004). The highest pixel of the image will locate at the peak and the lowest pixels will locate at the valley in the histogram (B.Dutta Majumder, June, 2011). Also, Felzenszwalb and Huttenlocher (2003) said that graph segmentation is to find the minimum cuts in the graph.

The light intensity shows in an grayscale image is measured in pixel and it has bit depth of 8 bit,16 bit and 32bits (Relf, 2004). For a bit depth of eight (eight bit), the intensity extremities contain 0 until 255 to represent dark until light as the shades of gray (Hossein Mirzabe et al., 2017). The researchers manipulated the non-zero values to 255 with a bit of eight to reduce the problem of having high background pixels in image (Durand-Petiteville et al., 2017). Higher bit requires larger memory amount and hence the bit depth of the image should choosse according the image level required to archive appropriate imaging solution (Relf, 2004).

Application of thresholding is based on gray level histogram to define local thresholding for every single sub-regions (Pal & Pal, 1993). Application of thresholding is based on gray level histogram to define local thresholding for every single sub-regions (Pal & Pal, 1993). Ideally, a bimodal histogram can ease the proses of thresholding (B.Dutta Majumder, June, 2011). An author for a book described every different image require different algorithm (B.Dutta Majumder, June, 2011). Felzenszwalb and Huttenlocher (2003) described two not similar type of local neighborhoods in constructing the histogram can produce better image segmentation using the correct algorithm. Image threshold is to classify pixels into dark pixel for object and light pixel for background (Pal & Pal, 1993). Another study also mentions that thresholding transforms the original image into bilevel thresholding, which consist of black object and white background (Chitradevi & Srimanthi, 2014). Thresholding cannot function perfectly for an image with multiple distinct gray level value for every object respectively but only two different gray level value, which are the object and background (N. Sharma et al., 2010). This statement is further support by a study shows that thresholding technique can work optimally when there is only two opposite pixels, which are the gray scale levels in the image (Fan et al., 2001). However, B.Dutta Majumder (June 2011) highlighted that normally a histogram for an image won’t form the ideally bimodal even though the situation is in acceptable situation, which has only two different regions shown in the image and hence a higher advanced algorithm is needed to add in the procedure of computing the grayscale histogram.

Figure 2.4 shows an example of binary set for a segmented image. S(x,y) is the resulting pixel at the coordinate (x,y) and the gray level is g(x,y) as well as T(x,y) is the threshold for the mapping gray scale {0,1} (Chitradevi & Srimanthi, 2014).

$$S(x, y) = \begin{cases} 0, & if \quad g(x, y) < T(x, y) \\ 1, & if \quad g(x, y) \geq T(x, y) \end{cases}$$

Figure 2.4: An example of binary set of segmented image (Chitradevi & Srimanthi, 2014)

To perform extracting object using grayscale segmentation, the image must be converted to grayscale image (B.Dutta Majumder, June, 2011). The hand region of images were having noise and then it was tranformed to grayscale image before converted into binary image using threshold (S. Sharma et al., 2015). Figure 2.5 shows the workflow of eight steps for hand shape and geometrical feature extraction which started from capturing a hand image until feature extraction.

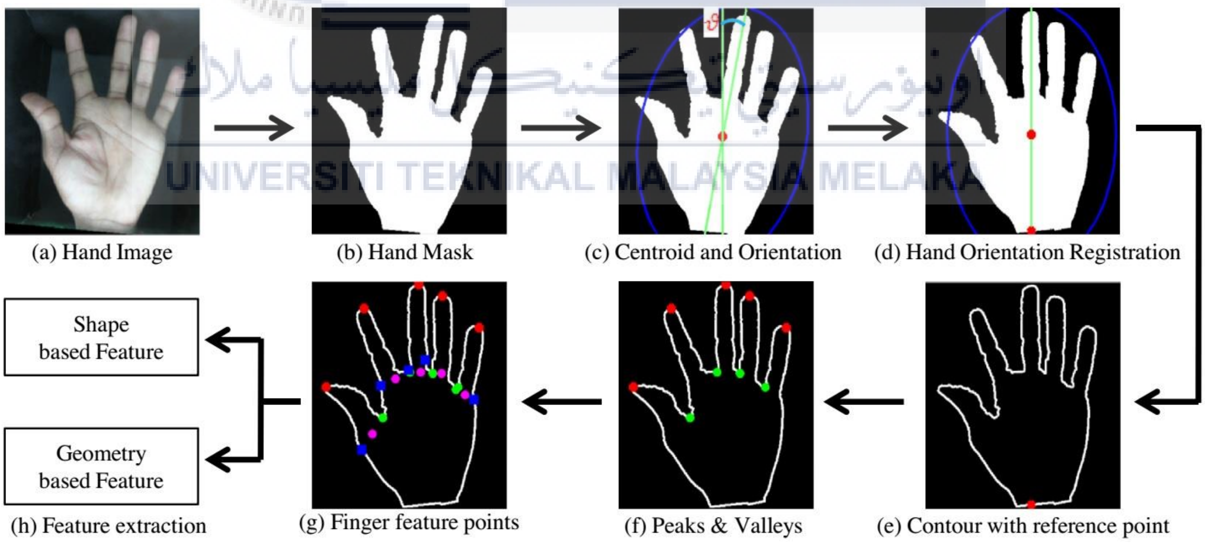


Figure 2.5: Workflow for hand shape and geometrical feature extraction (S. Sharma et al., 2015).

Zhu, Khoramshahi et al. (2015) hightlighted the image captured using amateur camera during day at the side of the bridge was in red green blue (RGB) form and it had to converted to gray image. Figure 2.6 shows the simple steps how to measure the traffic queue length from a original colourful picture to gray image to removed objects outside the road image to extraction highlights of the vehicle to calculate the queue length for each lane.

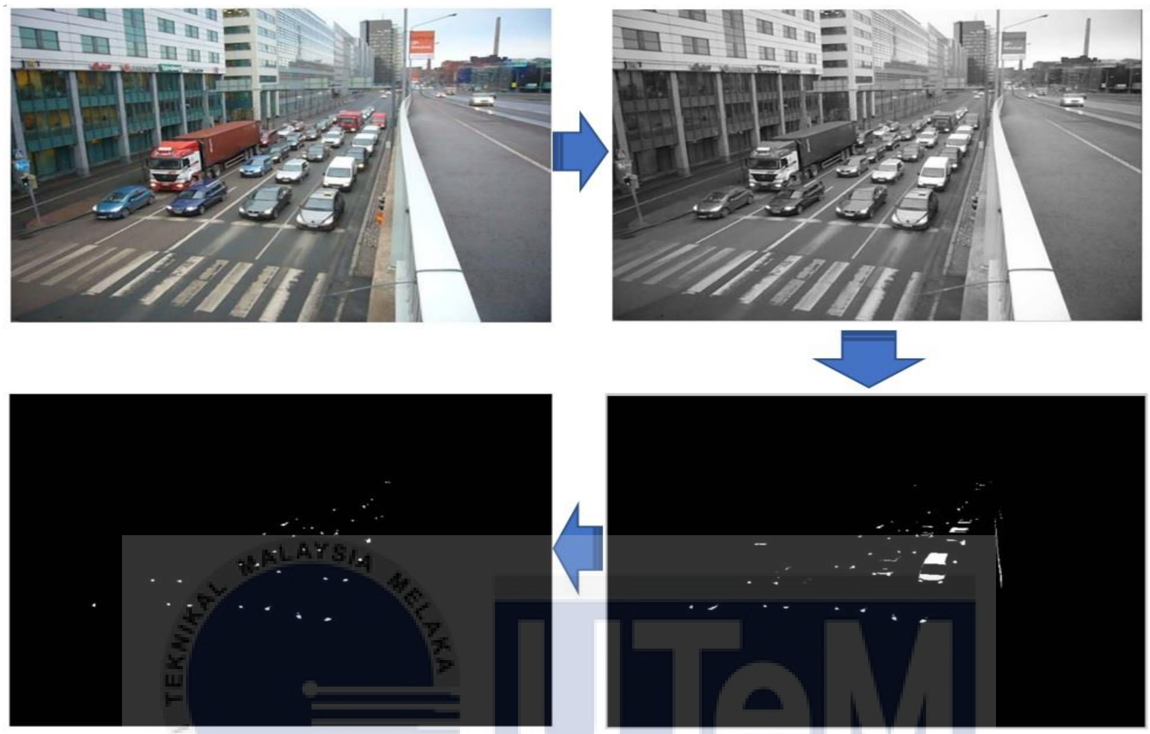


Figure 2.6: Workflow for traffic queue line segmentation (Zhu et al., 2015)

Figure 2.7: Original image (left), binary flesh image (middle), binary calyx image (right) (Durand-Petiteville et al., 2017)

Figure 2.7 shows the original image of a strawberry, its binary flesh image and binary calyx image. This strawberry agriculture study mentioned to transform the original image to gray level image is the initial step to extract the flesh and the whole fruit (Durand-Petiteville et al., 2017). Another study described after using the camera to capture an original RGB type image of the cucumber seed, it was converted into a grayscale with eight-bit intensity (Hossein Mirzabe et al., 2017). The workflow of the cucumber seed and kernel image processing is shown in figure 2.8.

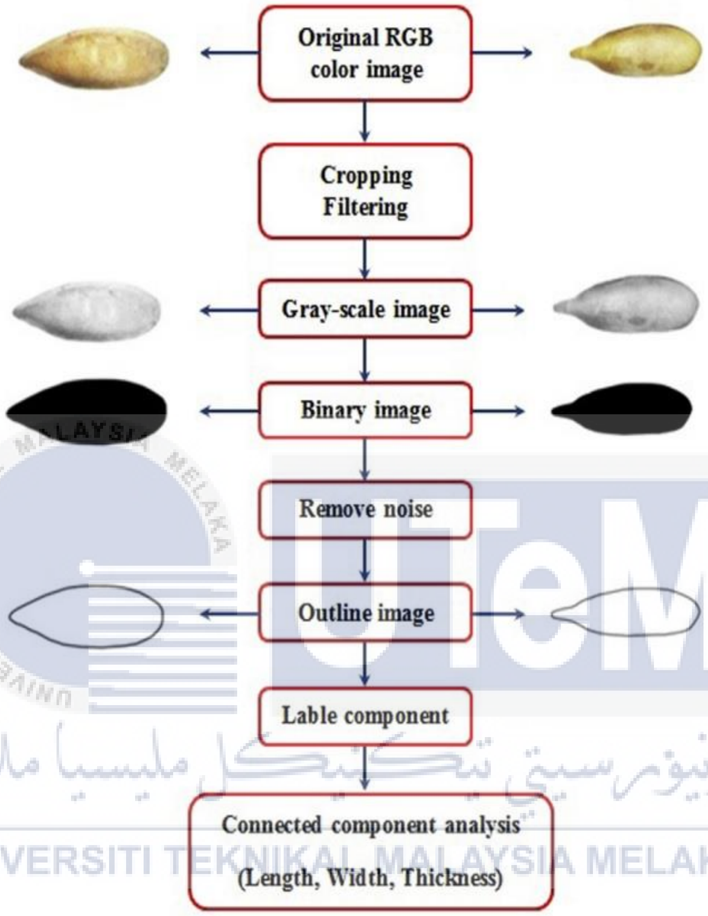


Figure 2.8: Workflow to measure the cucumber seed (left) and kernel (right) (Hossein Mirzabe et al., 2017)

Grayscale image segmentation is the most widely used. During preprocessing step, generating gray scale image from original image is the first step to eliminate the noise involved in the image (S. Sharma et al., 2015). In medical application, gray scale histogram thresholding method is largely used in TI weighted MRI brain image in medical by selecting the optimal peaks in histograms (N. Sharma et al., 2010). This statement is further support by Sharma and M. Aggarwal (2009) who described gray scale segmentation using histogram features is applied to gain MRI and CT images in medical field such as CT scan for abdomen image. Gray scale segmentation is used to segment the potential vehicle by first segment the road and then the potential vehicle from the image (Chitradevi & Srimanthi, 2014). Besides, a study shows using the method of thresholding on gray scale image, the area outside the road in image can be extracted (Zhu et al., 2015). The function of histogram is useful because it helps to identify the peaks and valleys of the fingers and the composite valleys for every peak to extract the hand of a gray scale image (S. Sharma et al., 2015). Durand-Petiteville et al. (2017) described a strawberry segmentation process by converting the color strawberry image to gray level image to extract flesh.

2.4.2 Image (color) segmentation

According to B.Dutta Majumder (June 2011), author for the book "Digital Image Processing and Analysis", color segmentation can be considered as the extended technique of the grayscale segmentation based on 4 reasons,

1. Both segmentation techniques manipulate color to proceed segmentation. Grayscale segmentation converts the original color image to gray level image. It is optional to proceed convertion from gray level image to binary image to ease the extraction. Color segmentation transforms the original image to color triple, which are red, green and blue (RGB).
2. Grayscale and color segmentation techniques use the behavior of the not equal pixels. The pixels are either intensity, color, motion, location or local attribute linked by edge in image. Unlike color segmentation, grayscale needs only to taking care of color information at 2 different pixels. Color segmentation focuses on 3 dissimilarity of pixels in an approriate manner.
3. The segmentation techniques extracted object based on the pixel information shows in histogram. Grayscale segmentation is using 2D histogram because it only consists of 2 different pixels information. However, colour segmentation has colour triple and hence it is using 3D histogram.
4. Both segmentation requires threshold approach to extract the image using the data shown in the histogram. For grayscale segmentation, the threshold approach is completely based on the pixels shown in histogram. Color segmentation does the same application but a further clustering technique will be first applied to the 3D

histogram because it needs to identify the color triplets respectively which represent in different region. After the application of clustering technique, threshold approach is applied.

Relf (2004) claimed that there are only two types of color image which are Red-Green-Blue (RBG) and Hue-Saturation-Luminance (HSL). The channel intensity extremities are from 0 to 255 for both type. Red, green and blue are the primary colors (Cheng et al., 2001). Color is more useful in computer vision because human beings treat color as a very essential thing in their visual experience (Engineering, 1990). A study stated that color segmentation is better than gray level segemntation because it provides more information (Cheng et al., 2001). Application of scale-space filtering need to use to know how many region for thresholding on a color triple histogram (Engineering, 1990). RGB color space can be presented using a 3-dimensional cube (Cheng et al., 2001). Figure 2.9 shows the 3-dimensional cube.

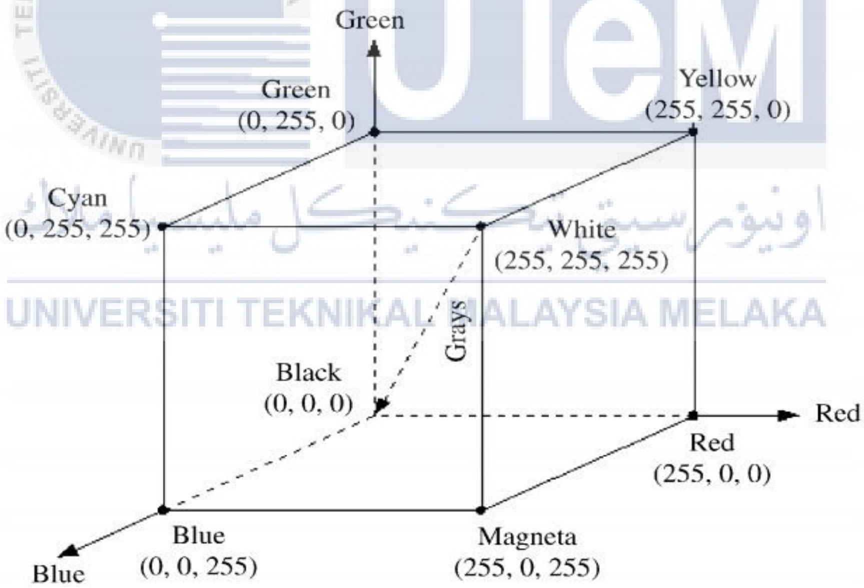


Figure 2.9: RGB in 3-Dimensional cube (Cheng et al., 2001).

Uhm et al. (2014) claimed that can RGB image processing method can apply to human body measurement for apparel application. In plantation industry, a study highlighted RGB segmentation approach can replace the manual process of grading the mango fruits based on the geometry mass (Momin et al., 2017). Figure 2.10 shows the workflow of the mango grading algorithm using color segmentation.

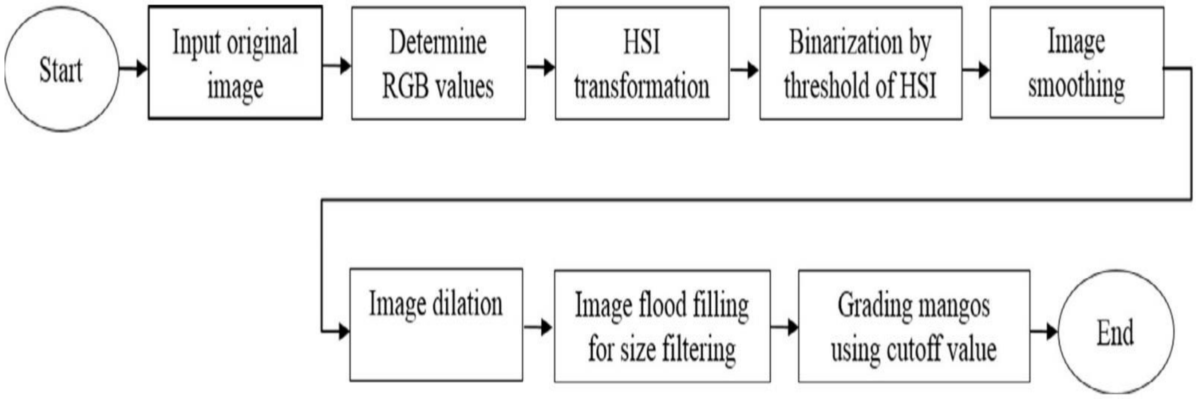


Figure 2.10: Workflow of the mango grading algorithm (Momin et al., 2017)

Application of color triple computer vision can use to measure the cover of vegetable to determine the condition of agriculture, for example to estimate the yield and agricultural loss by capturing the images during different period and investigate the RGB color intensity changes (Kırcı et al., 2014). Figure 2.11 shows the images of the field taken during different period.

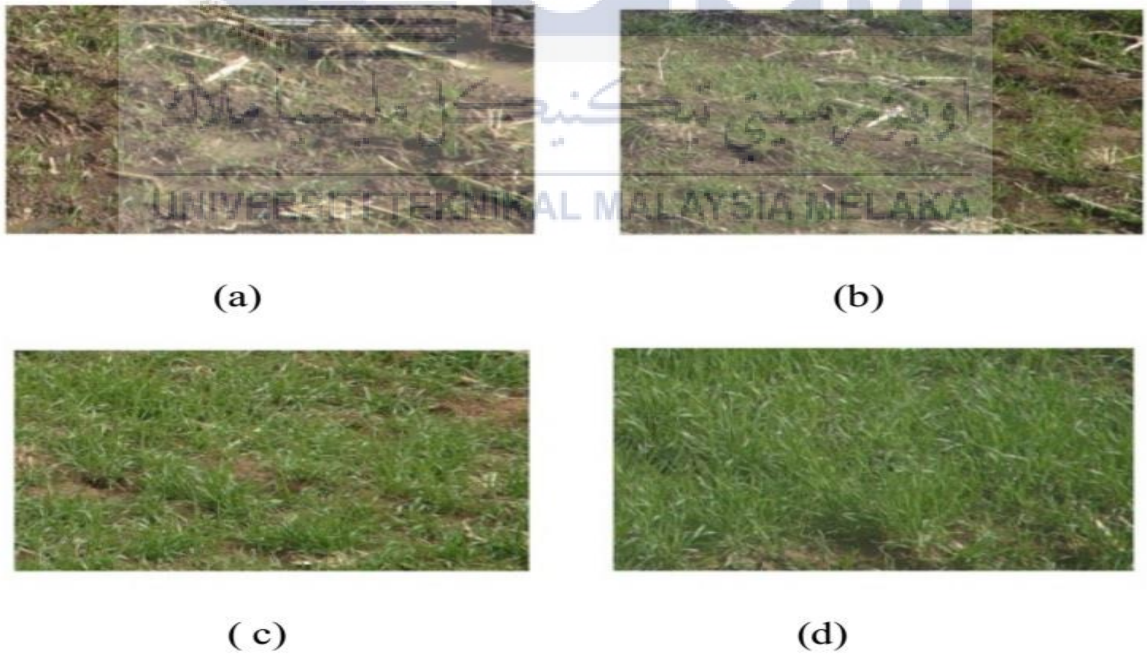


Figure 2.11: Images of field taken during (a) 25th Dec, (b) 15th Jan, (c) 13rd Jan, (d) 4th Mac (Kırcı et al., 2014).

2.4.3 Edge detection and segmentation

Edges also known as contours of image in the world of image processing and this approach contribute a better understanding for the algorithms of the vision computer to run and it provides a big leap towards human picture recognition system (B.Dutta Majumder, June, 2011). Edge and line segmentation can apply on the colour image, which is known as colour-edge extraction or it can apply on grayscale image and it will be called as grayscale edge extraction (Fan et al., 2001). Senthilkumaran and Rajesh (2009) said that an edge has strong relevant relationship with discontinuity image intensity or the first derivative if the image intensity. Dharampai and Vikram Mutneja (2015) mentioned that edge detection applies to detect the lines which present the limit and separate the object from another object or places in an image. B.Dutta Majumder (June 2011) provided that edge is the feature which divided two different regions based on the difference in colour or gray level pixels within a specific neighbourhood. The definition of edge detection is further explain as colour edges is the different neighbouring pixels of colours where the respective brightness values are very close (Fan et al., 2001).

Jiang and Bunke (1998) highlighted edge detection can work on simple control structures and regular operators as well as it is able to detect the region boundaries correctly and precisely. Just like the gray level and color segmentation, it is almost impossible to generate an edge detection algorithm which can be applied in different contexts and picture during the edge detection stage (Tabbone et al., n.d.). The strong point of using edge segmentation is it does not limit by the contextual information and hence they are considered flexible (Tabbone et al., n.d.). Ma and Manjunath (1997) claimed that in edge detection, there are two essential criteria to be taking care which are detecting and combining edges at more than one resolution.

The method of edge detection is classified into two categories, first is derivative approach and the second is pattern fitting approach (B.Dutta Majumder, June, 2011). Dharampai and Vikram Mutneja (2015) used another term to name the derivative approach and pattern fitting approach and they are gradient based edge detection and laplacian based edge detection. A study described gradient based edge detection as the edge pixels are detected using gradient method by defining the maximum and minimum in the first derivative followed by thresholding in the image and laplacian based edge detection as to detect the edges through the method of identify the zero crossings in the second derivatives (Maini, n.d.).

The performance of edge detection can be present using seven techniques, which are Sobel, Roberts, Canny, Laplacian, Kiresh and Edge Maximum Technique (EMT) as well as Prewitt (Al-amri, 2010). Figure 2.12 illustrate the seven techniques of edge detection.

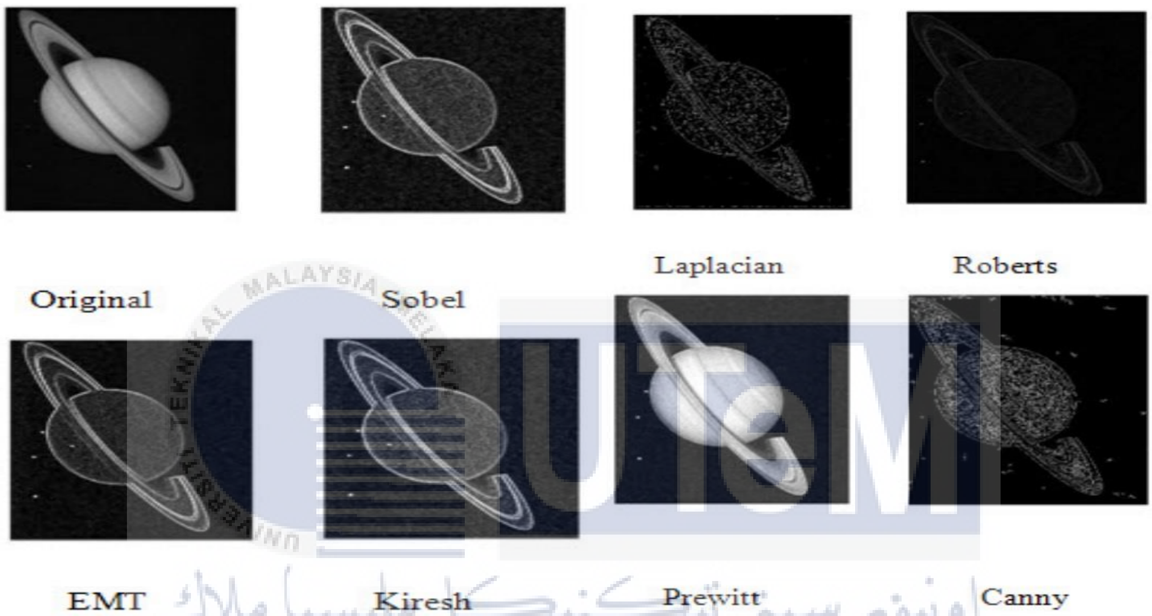


Figure 2.12: Seven techniques of edge detection (Al-amri, 2010).

Furthermore, Raman Maini and Dr. Himanshu Aggarwal (n.d.) investigated the advantage and disadvantage of some edge detection techniques to define the problem definition of the false edge detection. The edge detection techniques they investigated are Sobel operator, Robert's cross operator, Prewitt's operator, Laplacian of Gaussian and Canny edge detection algorithm which consists of six steps.

Basically, before the application of edge detection, a flow chart will first produce. Based on a report written by Dharampal and Vikram Mutneja (2015), the flow chart for edge detection usually consists of five steps. Figure 2.13 shows the steps described by Dharampal and Vikram Mutneja (2015).

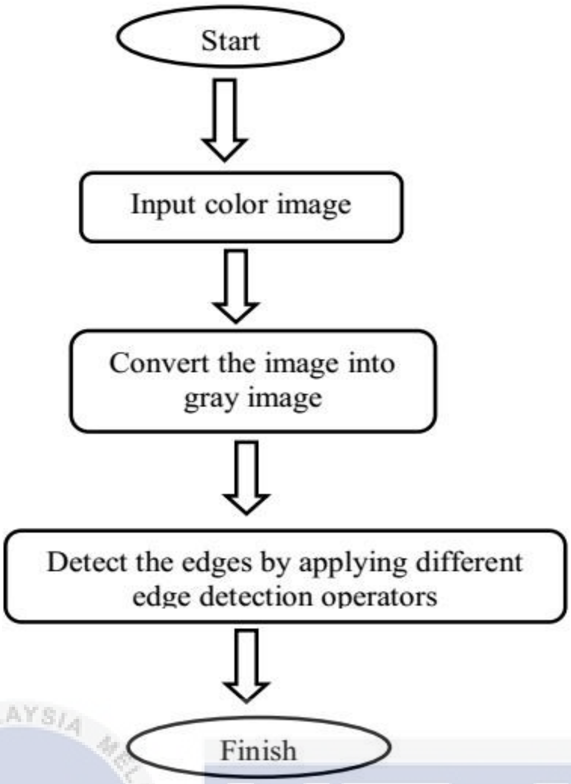


Figure 2.13: Flow chart for edge detection (Mutneja, 2015)

Edge detection approaches in image processing are widely used in industry, agriculture, plantation, inspection and anthropometry measurement. A research shows that edge detection and segmentation can applied in detecting the screw thread parameter with lower cost and higher efficiency (Min & Principle, 2015). For inspection purpose, E.S. Gadelmawla (2016) suggested a new technique by implemented a vision system and developed algorithms to process image processing and using edge detection and segmentation method to sort the pixels to calculate the features of a screw thread to take an inspection decision. It is impossible to ensure the dimensions and the axis of straightness of hot forgings, but with the application of image processing and edge detection, this measurement becomes possible (Zatočilová et al., 2016). Edge and line detection and segmentation also can be used to measure the glass micropipette tip geometry by implement a vision system. In anthropometry field, Klonowski et al. (2017) highlighted edge detection and segmentation technique can perform perfectly to become a tool for bioidentification due to its high precision and efficiency as well as low cost. Figure 2.14 shows the main features extracted from the image using edge detection and segmentation.

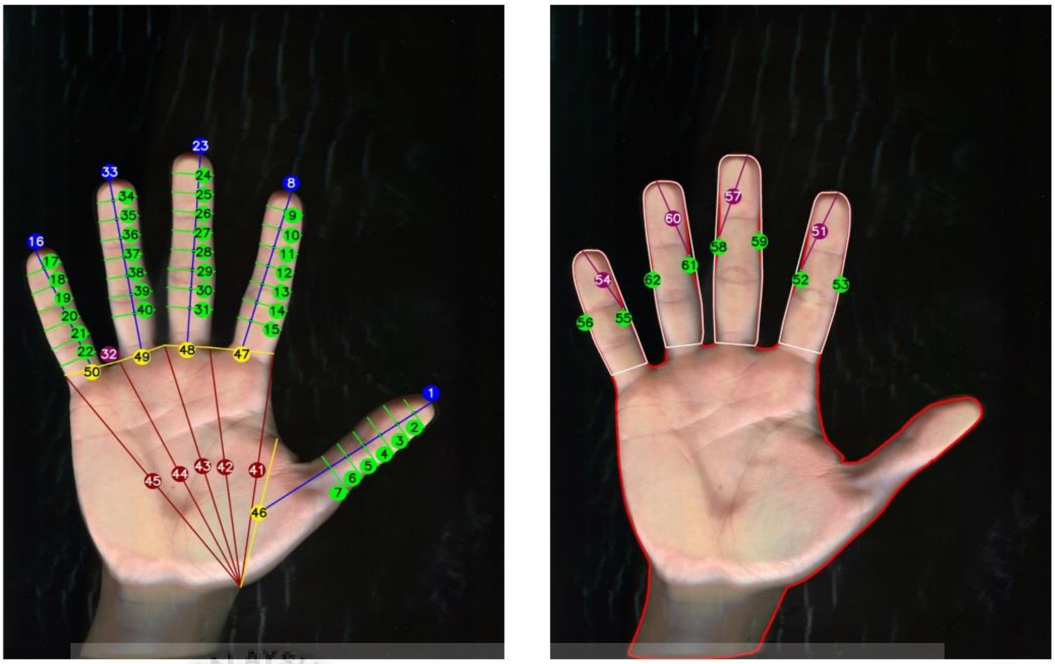


Figure 2.14: Main features extracted from the image using edge detection and segmentation (Klonowski, Plata, & Syga, 2018)

A study shows that the quality of rice grain can be identify by using edge detection in optical and image processing technique (Parveen, 2017). Figure 2.15 shows the edge detection and segmentation process to determine the chalky in the grain (rice).

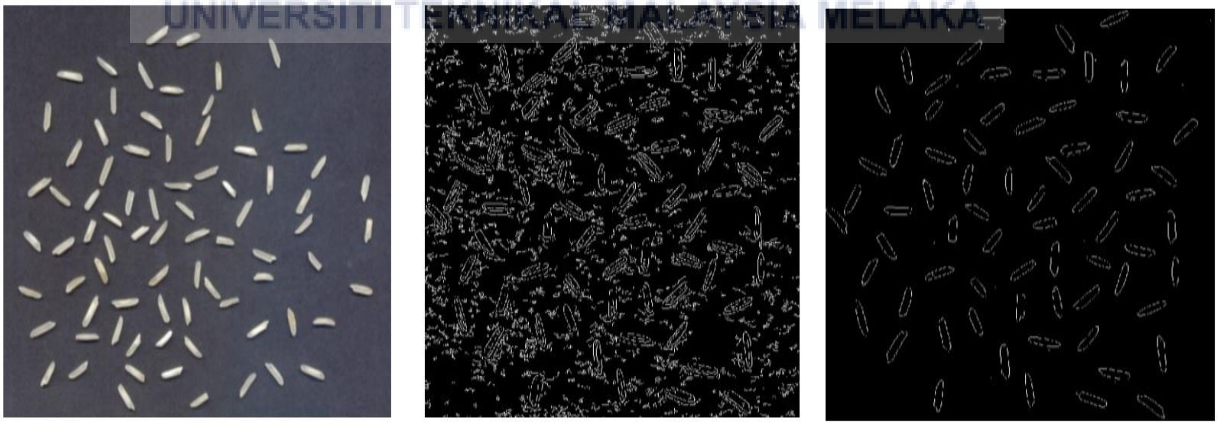


Figure 2.15: Process to determine the chalky in the grain (rice) (Parveen, 2017).

2.5 JOURNAL REVIEW

To carry out this project with better understanding and wide knowledge, a largely journal review had been done and it was followed by a deep filtering process to ensure the content was relevant to the project. A total of 68 selected journals and papers were used to analyze and as a supported evidence to write this chapter. The journals were divided into different categories to ease the process of written the whole chapter 2. Summarized of all the journals are shown in Appendix A.

2.5.1 Journal review table

In the journal review table, the information such as title of journal, author, publication year, parameter used, classified, application and remarks will be listed. This table is used to develop the journal mapping. All the information in the journal review table can be a guidance for the researcher. This table can be found in Appendix B.

All the journals written by previous researchers have been distributed according their categories and this classification is done using the parameter. Table 2.1 shows the summary of the classification of journals and papers used in 70 journals.

Table 2.1: Summary of the parameter used in 70 journals

Classification	Journals
Manual anthropometric measurement	11
Vision anthropometric measurement	10
3D gesture and image processing technique	7
Grayscale	13
Color	8
Edge and line	17
Computer vision method based on Python	4
Total	70

2.6 SUMMARY

In conclusion, chapter 2, literature review is all about the previous research and there is a total of five sub topic in this chapter. Firstly, an introduction for this chapter and it is followed by manual measurement and then vison measurement. Next, it is followed by digital image processing technique and lastly journal analysis. Only manual measurement, digital image processing technique and journal analysis contain sub topic. For manual measurement, there is only two sub topics, which are measuring instruments and issues of manual measuring. For digital image processing technique, it has three sub topics. They are image (grayscale) segmentation, image (color) segmentation and edge detection and segmentation. For journal review, it consists of journal review table and its description.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The definition of methodology is the study of organized the work flow using the correct methods. Generally, the whole idea of writing methodology is to illustrate the idea how the hand measurement using image processing can be implement step by step. Hence, in this chapter, every single step includes in the procedure will be outlined. Every method used in this chapter will also be discussed. To simplify the sequence of execution and the implementation of this project, flowchart is used to replace the long sentence words to save time understanding the whole sequence. Every sequence will have subsequent detailed explanation aims to provide guideline and deeper understanding as well as for as a guidance for duplication the flow of this project.

3.2 OVERVIEW OF METHODOLOGY

Gantt chart acts as a schedule to control the time frame and it can be found in Appendix C and D which the work flow shown is within 15 weeks period. There is a total of 7 stages to carry out the whole project. These 7 stages are in planning stage and hence the result of every single stage can become the factor of making the decision of using a replacement to change the original method. Every single stage has their own intent and target to achieve. A good target achievement can translate as the desired reason from that method is attained. Only then design of experiment develop followed by system structure of this project. Figure 3.1 shows the task to be taken in step by step to develop a vision measurement for hand parameter using a flow chart.

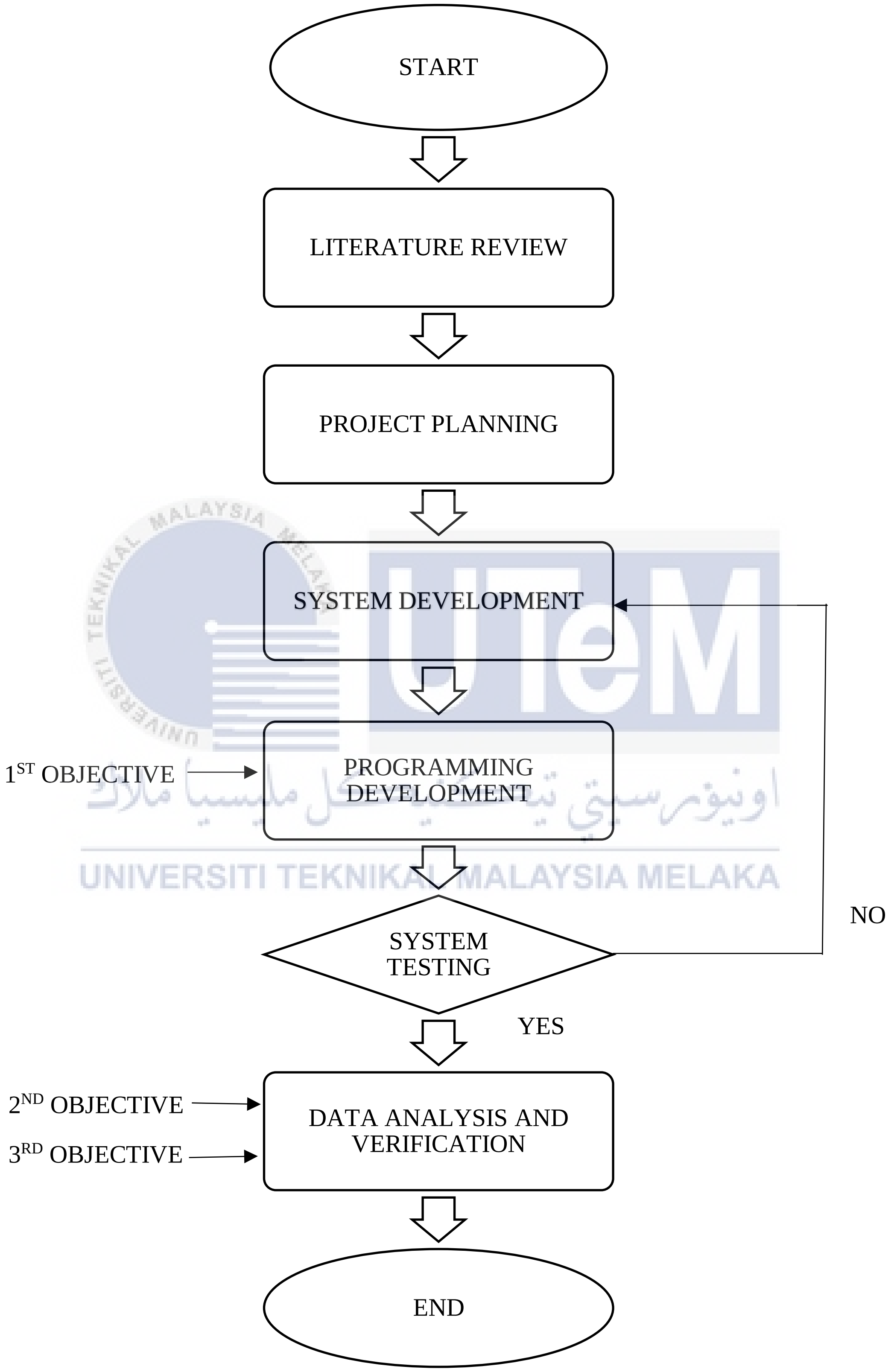


Figure 3.1: Flow chart of the vision based hand measuring procedure

3.3 LITERATURE REVIEW

In short, literature review is all about the theory and previous research from the other researchers. All the essential and relevant information from previous studies are extracted as guideline and reference. This is the very first step when carry out this project because a deeper understanding required and by doing a literature review, many guideline, suggestion and reference can become a tool to proceed another step.

Mostly, previous research used for literature review are journals and minority are from books. All these journals downloaded from internet website such as IEEE and Science Direct. This journal online searching process were aided using the UTeM library website to shorter the time duration. Then, it was followed by a filtering process. This filtering process aims to keep only the quality and official journals as well as relevant to this project. All the journals which were not passed will be rejected. The passed journals were tabulated and listed out title, year, author, objective, parameter used and application. Afterwards, the journals were classified based on their parameter to write the literature review. Books used for this project also were borrowed from UTeM library by using the library online searching machine to search for the relevant material.

The computer software Microsoft words was use for writing and grouping literature review. The original format can be modified to official format such as margins, font type and font size as well as spacing in this software within seconds.

3.4 PROJECT PLANNING

Project planning is a primary idea how to perform method to solve this research problem statement and complete the objectives mentioned. Besides, during the stage of project planning, it needs to cover the three scopes of the research. First, Python programming will be involved and discussed in detail in programming development stage. Second scope is about the hand measurement. Hand measurement only includes finger length, palm length, palm breadth and finger vs palm (shape) index will be calculated. The limitation of image processing is it cannot measure finger vs palm index directly from the image. Due to this reason, an additional formula will be inserted to the script to ensure it is automatically calculate based on the measurements taken from image. The last scope is collecting 30 hand data. A total of 30 subjects will be randomly chosen to have his/her right hand measured using Vernier caliper for manual measurement and developed vision measuring system. All the subjects must be Asian for standardizing data purpose. The way how these hand measurements collected will be discussed in detail in programming development stage for vision measuring method and in data analysis and verification stage for manual measuring method.

Since this system includes hardware and software parts, it will be explained separately in system development to clarify the hardware and software used to develop the whole system. Then, it will be followed by writing the programming script for the system to run. The output of system development and programming development will be tested to ensure the environment is stable and it will be discussed in detail in system testing. When the system testing has completed achieve its minimum required, the system will be used to collect raw data. Compare the result in terms of time taken and data of hand measurement with manual measurement. Due to the project time constrain, the data for manual and vision measurements will be collected before the fully development of vision measuring system and afterwards using offline method to attain the measurement.

3.5 SYSTEM DEVELOPMENT

An automatic system cannot stand alone with the programming part. In short, this research of vision based hand measurement system contains a hardware and a software system. Hardware system consists only physical touchable elements such as camera and controller. They usually connect to each other using cables, wires or even advanced sensor to detect. Software system is the programming generated system functions to arrange tasks in an ascending order way and assign tasks one by one to every individual hardware to perform. It means that there will have a procedure scripts for the system to perform from step 1 until last step. The communication between hardware and software systems ensures the interfacing task completing between them. Besides, software is also the main core help to reduce the defects generated by the hardware system using library installed within the system in controller. For example, a blur picture captured using camera will be sharpen using the well-written programming.

3.5.1 Hardware system

A hardware system means all the components work together to produce a desired output. It will have a systematic sequence for the components to perform the individual task. All the information transfers in the form of signal. The signal transmits from the camera to the controller and display on laptop screen. Basically, this project hardware system consists of three components which are a controller, a camera and a laptop and they connected using cable. Table 3.1 shows hardware components and description.

Table 3.1: Hardware components and description

Hardware	**Description**
Laptop	A laptop is used as a screen to display viewing and to provide mouse and keyboard for user input function.
Controller	It is a mother board on a single integrated circuit and acts as microprocessor core, memory storage and programmable input or output peripherals to another electronic component.
Camera	To capture an image for image processing purpose.

Controller is a communication medium between camera and laptop. It receives the image signal transmitted from the camera and runs the image processing within itself. Laptop functions to display the internal environment of controller but the image processing will only run within the controller. This totally separated environment of laptop and controller can avoid file crashing and system down in laptop because laptop has its own system to run at its background to support all the display functions. Camera functions as the vision sensor to capture picture once the order is received. Figure 3.2 depicts the flowchart of hardware system and the arrows show the signal transfers from one device to another device.

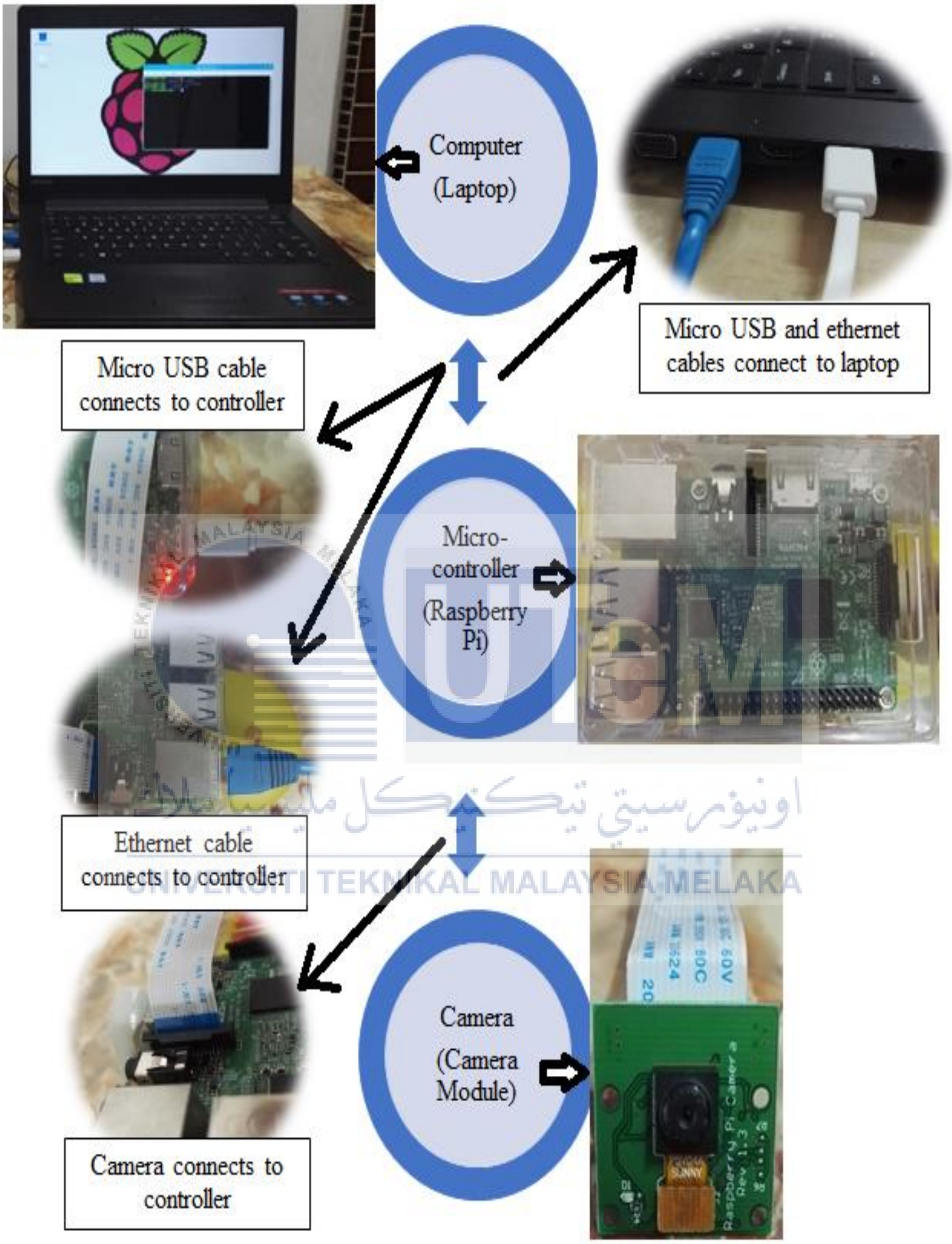


Figure 3.2: Flowchart of the hardware system

Having understand how these electronics devices integrated brings benefit towards searching for the suitable candidate. Then, market research was done to have a deep background study on target candidates to ensure the flow of the process is smooth and communication between one device to another device does not have barrier.

Raspberry Pi (RPi) is used as the controller because it is capable to support Python language for writing image processing script. Besides, it has its own Raspberry camera module which can directly connect to the controller. Raspberry camera module can support the compatible of camera and controller communication. This is due to the reason both controller and camera are from the same manufacture and company. This feature ease the software system to integrate and manipulate the functionality of the camera in receiving image data. There is another attractive benefit which is the price for a single Picamera is very cheap and the size is small as well as it is light. Hence, it also eases the camera setup procedure later. Figure 3.3 shows example of Raspberry Pi and Raspberry camera module.

Figure 3.3: Raspberry Pi and Raspberry camera module
(https://projects.raspberrypi.org/en/projects/getting-started-with-picamera)

For laptop, Lenovo brand with 64-bit environment is used. Table 3.2 describes the characteristics of the Lenovo laptop, Raspberry Pi and Raspberry camera module.

Table 3.2: The model of the Lenovo laptop, Raspberry Pi and Raspberry camera module.

Hardware	**Model**
Laptop	Lenovo Ideapad 310-14IKB
Controller	Raspberry Pi 3 Model B
Camera	Raspberry Pi Camera Board v1.3

3.5.1.1 Hardware system specifications and features

Generally, there is a total three different type of hardware using in this project and all the specifications and features for different hardware are listed in the table 3.3.

Table 3.3: Specifications and features

Lenovo Ideapad 310-14IKB	
CPU	64-bit
Processor	Intel i5 – 7200U 2.5G @ 2.50GHz & 2.70GHz
RAM	8GB
HDD	1TB
System Type	64-bit Operating System
LAN	100M/1000M
Display	14.0” FDH LED
Battery	2Cell
OS	Windows 10 Home SL
Raspberry Pi 3 Model B	
Environment	64-bit
SoC	BCM2837
CPU	Quad Core A53 @ 1.2GHz
RAM	1GB SDRAM
Instruction set	ARMv8-A
GPU	400MHz VideoCore IV
WiFi	BCM43438 WiFi 10/100 ethernet
Wireless	802.11n
Bluetooth	Bluetooth Low Energy (BLE) on board

GPIO Pin	40-pin extended GPIO
USB	4 x USB 2 ports
Port	4 pole stereo output and composite video port
Display	Full size HDMI
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
Camera	CSI camera port to connect a Raspberry Pi Camera
Storage	Micro SD
Raspberry Pi Camera Board v1.3	
Pixel	5 Megapixel
Omnivision	5647 Camera Module
Photo Pixels	2592 x 1944 pixels
Video Resolutions	-1080p@30fps -720p@60fps - 640 x 480p90 /90 recording
Size	20 x 25 x9mm
Weight	3g
Supported	Connect to all latest Raspbian Operating System
Compatible Model	Fully compatible with both Model A and Model B Raspberry Pi
Pins	15-pins MIPI Camera Serial Interface- Plugs Directly into the Raspberry Pi Board

3.5.2 Software system

Figure 3.4: Python logo
(https://realpython.com/learn/python-first-steps/)

Figure 3.4 shows a Python logo. Python version 3 is used as the default programming language to develop the coding and program it in controller to do the image processing.

Python is an interpreted high-level programming language first released in 1991 designed by Guido van Rossum. It is a code readability which supports multiple programming paradigms such as object-oriented, imperative, functional, procedural and standard library. Python also famous with its easy to read, fast and versatile features as well as it is an open source software with community-based development model to cover almost entirely variant implementations. Majority developers used Python with installed target library to do image processing, data analysis and games creating. The examples of organizations use Python are Wikipedia, Google, Yahoo, CERN, NASA, Facebook, Amazon, Instagram, Spotify and Reddit social news networking was fully written in Python.

3.5.3 Installation guide

Installation guide is divided into hardware installation and software installation. A complete set of installation will follow by integrating between hardware and software to connect each other and then transforming into a system. This section will provide the guideline and explain the steps in details.

3.5.3.1 Hardware installation guide

As discussed, the hardware consists of a laptop, controller and camera. However, to connect all hardware, it requires a connection medium. Raspberry Pi controller must be connected to internet to function and hence an Ethernet cable is required. Besides, Raspberry Pi controller needs a medium to store the data and library information, to run the system and to support the program and software such as LibreOffice. Table 3.4 shows extra parts needed.

Table 3.4: Extra parts needed

Number	Part Number	Description	Function
1	Micro USB cable	Also known as Android phone cable.	To connect controller to laptop.
2	Ethernet cable	Also known as Lan straight cable	To enable internet connection from laptop to controller.
3	Micro SD card	Preferable minimum 8GB	Storage purpose

4	Universal memory card reader	Preferable USB head and multi card port	To transfer data to the micro SD card.

Figure 3.5 shows the ports on Raspberry Pi controller. Ethernet cable, micro USB cable and micro SD card should connect and insert to 10/100 Lan port, micro USB power input and micro SD card slot respectively. The other ends of ethernet cable and micro USB cable are connect to their respectively port on laptop.

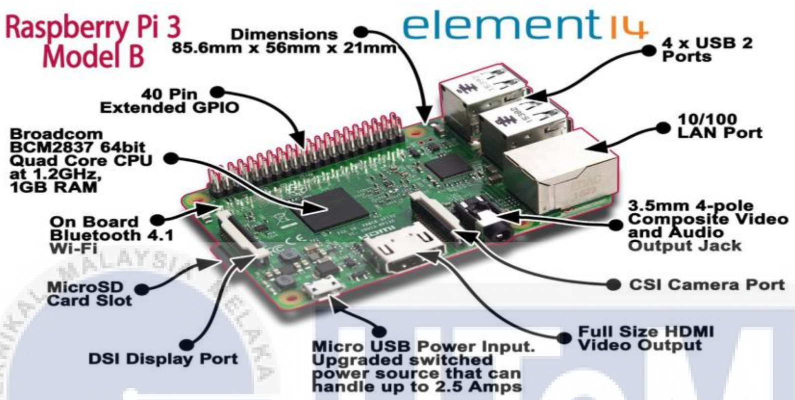


Figure 3.5: Ports on Raspberry Pi controller
(https://www.takealot.com/raspberry-pi-3-model-b-1gb-project-board/PLID41466406)

Raspberry Pi camera board v1.3 usually comes with its ribbon FFC line. It's the other end of ribbon FFC line to insert to CSI camera port by slowly push up the camera port holder and attach the flat ribbon before pushing down the holder. Figure 3.6 shows the Picamera installation guide.

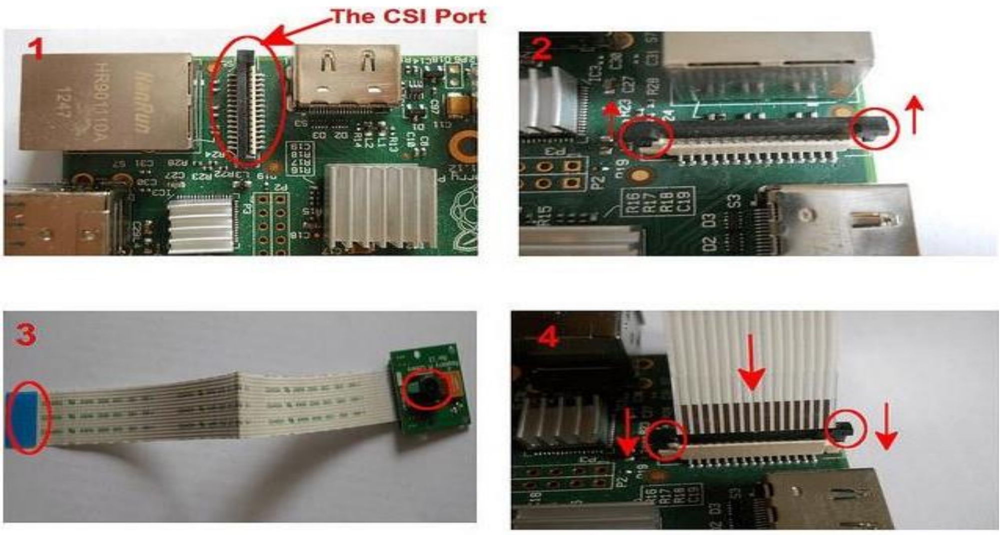


Figure 3.6: Picamera installation guide (Nora, 22nd Dec 2016)

3.5.3.2 Software installation guide

Software installation guide can be categorized into 2 parts. The first part only focuses on connection the hardware using laptop as a screen to display. Then, the second part will focus on steps how to install OpenCV and Python and compile all these required libraries and supporting files in controller.

To begin with, all the setup installers need to be downloaded. Table 3.5 below shows the list of installers with their function and installer website address.

Table 3.5: List of installers

Installer	Function	Installer Website Address
Raspbian Stretch with Desktop	Official supported Raspberry Pi operating system	https://www.raspberrypi.org/downloads/raspbian/
Etcher	Flash OS images to SD cards and USB drives	https://etcher.io/
Putty	Terminal emulator, serial console and network file transfer application.	https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
VNC Viewer	Provide remote access feature	https://www.realvnc.com/en/connect/download/viewer/
SD Memory Card Formatter	Format SD card while protected sensitive area	https://www.sdcard.org/downloads/formatter_4/

When all the installers have downloaded and installed in laptop, mount SD card to laptop by inserting the micro SD card to universal memory card reader and mount to laptop. Use SD memory card formatter to format the SD card. Remember to change the volume label before hit format. Open etcher software to flash the browse zip file of operating system (OS) of Raspbian Stretch to the SD card. Hit flash and wait until it finishes validating. Unmount SD card and remount it again. Open the “boot(F:/)” in computer and create a new text document and name it “ssh”. To remove file extension of the “ssh.txt” text document, just rename it to “ssh”. Unmount SD card and insert it to Raspberry Pi controller. Setup the hardware system using the cables to connect to laptop. Afterwards, open putty and type “pi” for *login as* and “raspberry” for *pi@raspberrypi.local* and hit enter. After typing “sudo raspi-config” for *pi@raspberrypi: ~$* and hit enter, select interfacing options and choose VNC to enable it. Figure 3.7 shows the interfacing options.

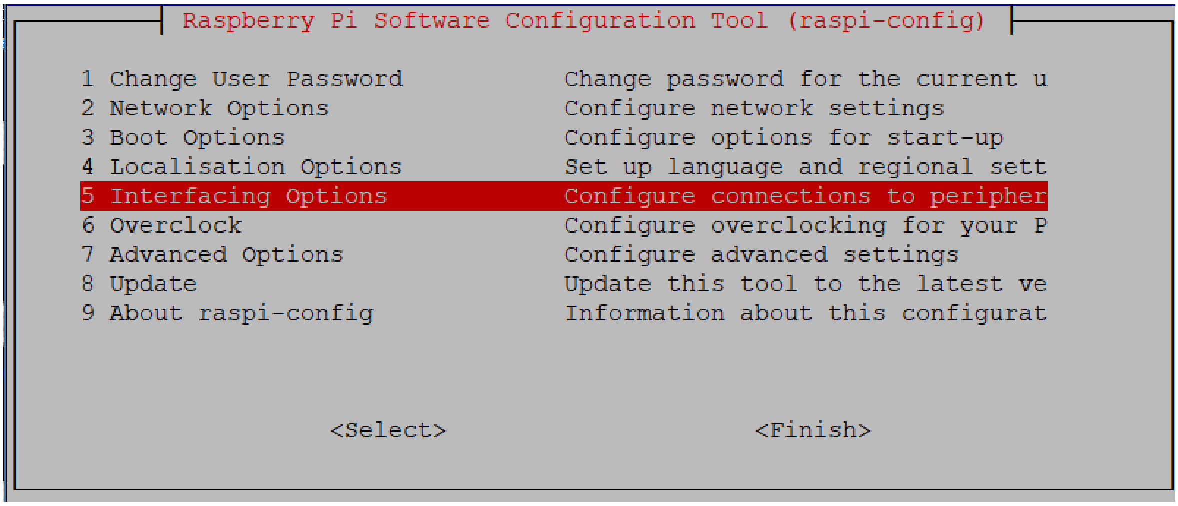


Figure 3.7: Interfacing options

Click finish on the display window and quit putty. Open VNC viewer software on laptop. Type "raspberrypi.local" on the search engine on VNC server screen and hit enter. An authentication will pop up. Change the username to pi and password to raspberry and hit enter. VNC viewer will load and a screen will display. This is the remote Raspberry Pi screen. Figure 3.8 shows remote Raspberry Pi screen.

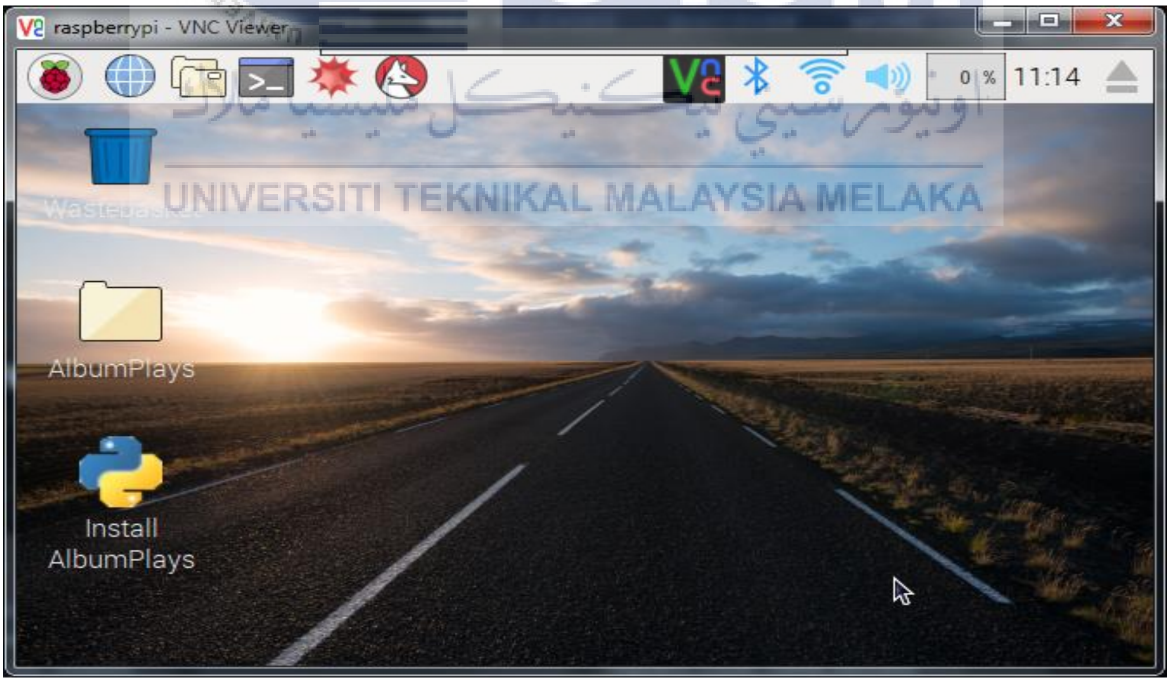


Figure 3.8: Remote Raspberry Pi screen

Having finish the part one of software installation guide, the screen of Raspbian is able to display using laptop. Then, the second part of software installation guide can begin to install all the necessary libraries and file extension for coding development. This part is going to take at least 6 hours to complete. Hence, ensure to lift up a corner of Raspberry Pi controller for ventilation purpose to prevent overheating and then CPU burn. All the commands for the following installation will type in the terminal on Raspberry Pi screen. Figure 3.9 shows the terminal on Raspberry Pi remote access screen.

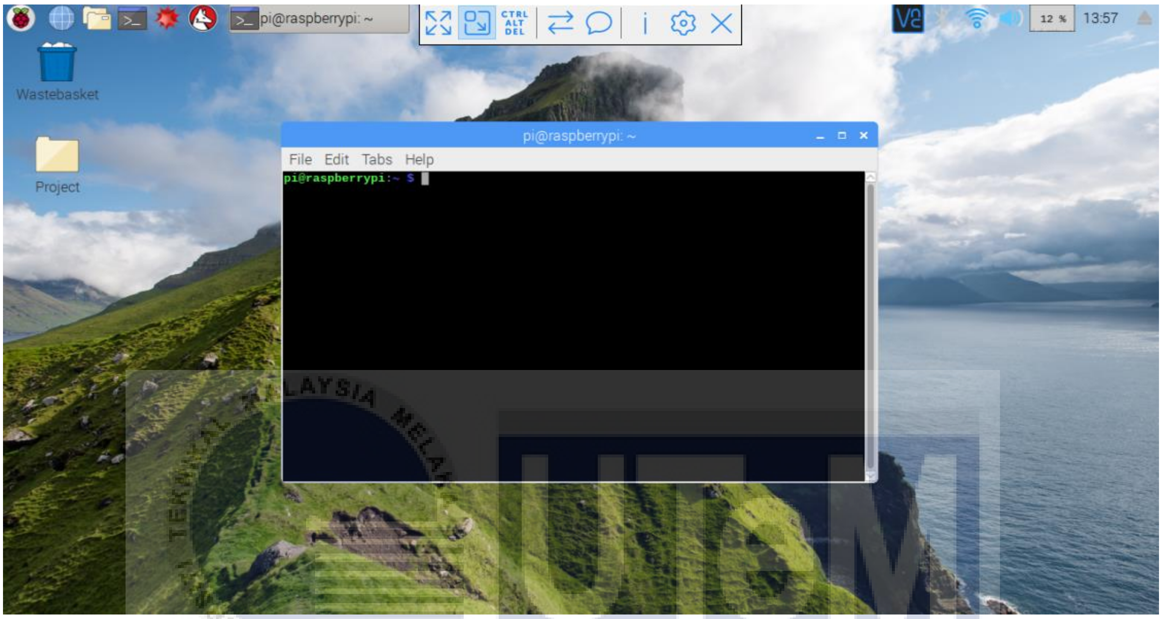


Figure 3.9: Terminal on Raspberry Pi remote access screen

Before start, expand filesystem is very important to include all available space on micro-SD card. This checking can be done by typing “sudo raspi-config” on the terminal and select “Advanced Options” followed by “Expand Filesystem” then hit finish after it is done expanding.

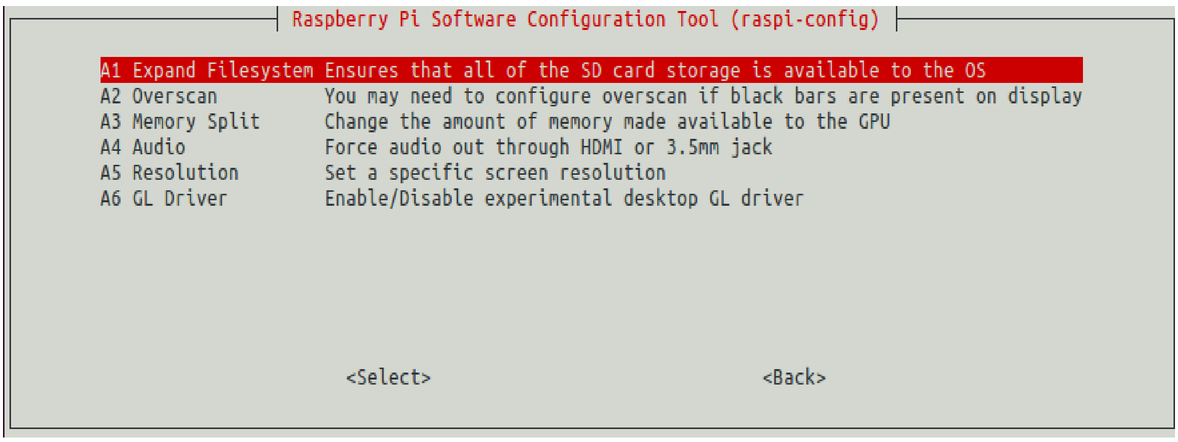


Figure 3.10: Expand filesystem configuration

The following are the commands to complete the whole installation. It is started with install dependencies, download the OpenCV source code, installing Python 2 and Python 3, virtualenv and virtualenvwrapper environment, installing NumPy and Matplotlib, examine the output of CMake, configure swap space size, compilation and install OpenCV, changing directory, testing installed OpenCV and lastly swap size back.

1. Install dependencies

   $ sudo apt-get update && sudo apt-get upgrade

   $ sudo apt-get install build-essential cmake pkg-config

   $ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

   $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

   $ sudo apt-get install libxvidcore-dev libx264-dev

   $ sudo apt-get install libgtk2.0-dev libgtk-3-dev

   $ sudo apt-get install libatlas-base-dev gfortran

   $ sudo apt-get install python2.7-dev python3-dev

2. Download the OpenCV source code

   $ cd ~

   $ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.4.1.zip

   $ unzip opencv.zip

   $ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.4.1.zip

   $ unzip opencv_contrib.zip

3. Installing Python 2 and Python 3

   -Using pip feature to install

   $ wget https://bootstrap.pypa.io/get-pip.py

   $ sudo python get-pip.py

   $ sudo python3 get-pip.py

4. Virtualenv and virtualenvwrapper environment

   $ sudo pip install virtualenv virtualenvwrapper

   $ sudo rm -rf ~/.cache/pip

-Type “nano ~/.profile” and add the following lines at the bottom of the file

virtualenv and virtualenvwrapper

export WORKON_HOME=$HOME/.virtualenvs

export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python2

source /usr/local/bin/virtualenvwrapper.sh

-Press Ctrl + x and y and enter

-To log back in virtualenv environement for checking

$ source ~/.profile

-To create Python 3 virtual environment

$ mkvirtualenv cv -p python3

-If there is a “(CV)” in front of pi@raspberrypi : ~$ means the virtual environment is successfully installed.

5. Installing NumPy and Matplotlib

   $ pip install numpy

   $ pip install matplotlib

6. Examine the output of CMake

   -Ensure the terminal is in virtual environment, then add the following commands

   $ cd ~/opencv-3.4.1/

   $ mkdir build

   $ cd build

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.4.1/modules \
    -D BUILD_EXAMPLES=ON ..
```

   -Search for Python 3 details and check if the virtual environment is installed.

   -Check the libraries and dev files

7. Configure swap space size

   $ sudo nano /etc/dphys-swapfile

-Change the default value “CONF_SWAPSIZE=100” to “CONF_SWAPSIZE=1024”

-Adding the following lines and stop and on the size configuration

$ sudo /etc/init.d/dphys-swapfile stop

$ sudo /etc/init.d/dphys-swapfile start

-To verify the amount of memory + swap, can add the following line

$ free -m

-The output should be like this

```
total     used     free   shared  buffers   cached
Mem:          435       56      379       0       3       16
-/+ buffers/cache:       35      399
Swap:        1023       0     1023
```

8. Compilation and install OpenCV

   -The first line commands

   $ sudo make

   $ sudo make install

   $ sudo ldconfig

9. Changing directory

   $ ls -l /usr/local/lib/python3.5/site-packages/

   $ cd /usr/local/lib/python3.5/site-packages/

   $ sudo mv cv2.cpython-35m-arm-linux-gnueabihf.so cv2.so

   $ cd ~

   $ cd ~/.virtualenvs/cv/lib/python3.5/site-packages/

   $ ln -s /usr/local/lib/python3.5/site-packages/cv2.so cv2.so

10. Testing installed OpenCV

    -Enter virtual environment

    $ cd ~

    $ source ~/.profile

    $ workon cv

    -Testing installed OpenCV

-Adding the following line will show the information of Python installed

$ python

-Adding the following line should give no error

>>> import cv2

-Adding the following line should response the version installed for OpenCV

>>> cv2.__version__

11. Swap size back

$ sudo nano /etc/dphys-swapfile

-Change the default value “CONF_SWAPSIZE=1024” to “CONF_SWAPSIZE=100”

-Adding the following lines and stop and on the size configuration

$ sudo /etc/init.d/dphys-swapfile stop

$ sudo /etc/init.d/dphys-swapfile start

The Python with OpenCV is finally done and it is ready to develop source code for image processing on hand measurement. Figure 3.11 shows verify the Python with OpenCV installation.

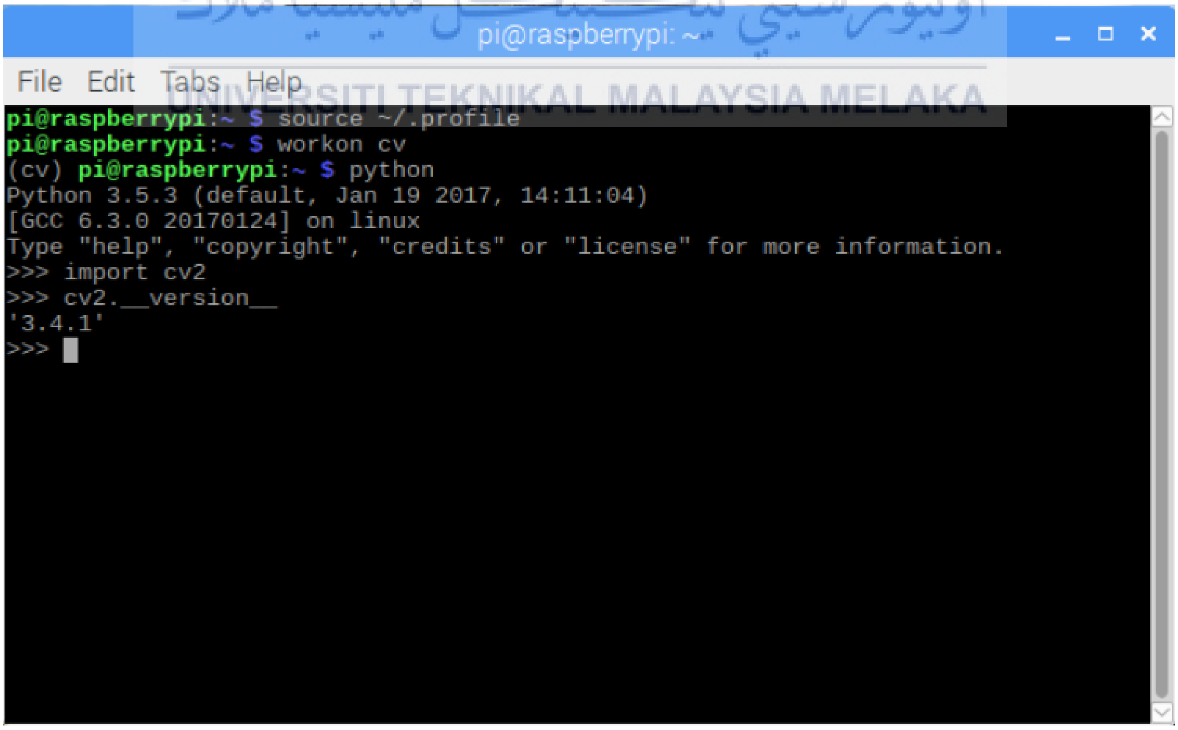


Figure 3.11: Verify the Python with OpenCV installation

3.6 PROGRAMMING DEVELOPMENT

A standardized sequence is needed. To develop a source code which able to carry out detection and measurement from a hand image, a sequence of image processing techniques must be identified. A draft of sequence will be generated first and went through filtering and rechecking logic to make the flow become more reliable. This is to ensure every single step on the sequence has its own functionality to perform the task and those non-function or non-essential steps will be eliminated to shorten the source code. A shorter source code can save the storage needed and time to run module. If the draft is too confusing, a flow chart can use to replace hand writing to show the linkage of the function and where to perform loop. Flow chart can save time for understanding and easier for modification.

It is an online-type programming. This means that after an image is captured, image processing techniques is processing and lastly display the measurement of the hand parameter as the output. It is a continuous flow without a break. Thus, Raspberry Pi camera module plays a significant role to start the writing script by initializing the camera to capture an image. The image data is being process using image processing technique. Lastly, hand parameter calculation and convert it to the real-world measurement. Basically, these 3 steps make the main functions. From capturing image as start, image processing as the body and parameter calculation as the end.

Define, def is widely use in this program. Within those 3 main functions, many small functions need to be added to perform its own task and a cumulative of all the small roots can make up the whole body of the source code. However, adding them line by line to the source code will make it become very long and hard to perform correctly. Even if the program can run, the whole source code will be very confusing and an occurred bug will be difficult to detect its root of problem to resolve it. In short, arrange the source code according to its def function then call them when needed within the main working procedure will be the best idea. It can drive to easier modifying and minimize the risk of confusing within the program itself. These 3 requirements will be used for writing scripts. Figure 3.12 shows the sequence of image processing technique.

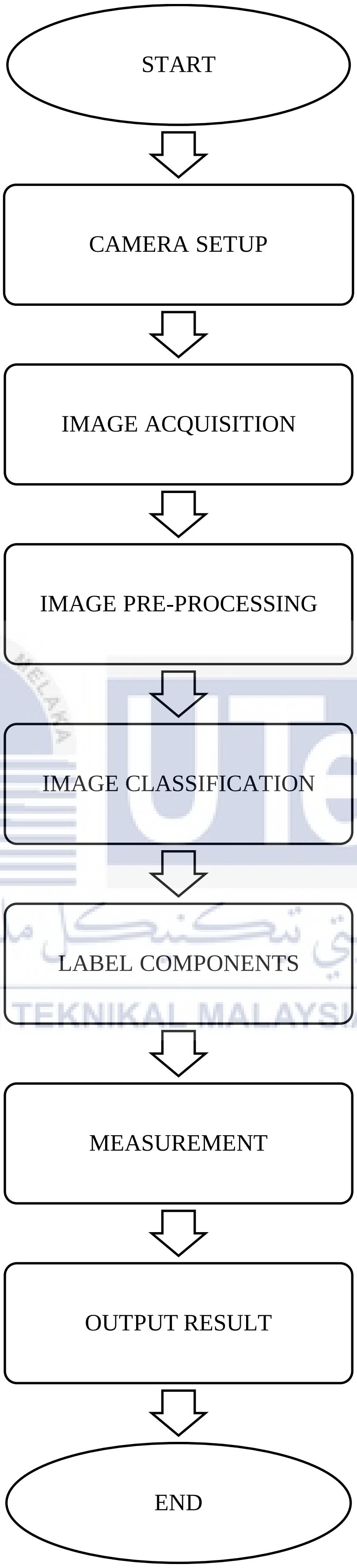


Figure 3.12: The sequence of image processing technique

3.6.1 Start

Start is the first step to write the source code. To begin with, libraries are included. Identify which library requires is very important to run the operation or else syntax error, run-time error or logic error will pop up. This is because the script consists of unknown variable or function if the specific library is not import in the first part. Python is very flexible because developers can modify the imported library at any time without constraint. Thus, it is not necessary to set which library to use in the beginning but ensure to import the library before calling its function. Table 3.6 shows some examples of essential library.

Table 3.6: Examples of essential library

Number	Example	Library
1	Import cv2	OpenCV version 2
2	Import numpy as np	Numpy
3	Import os	Operating system
4	Import imutils	Imutils
5	Import matplotlib	Matplotlib

As mentioned, this is an online-type programming. Hence, camera will be used to capture image and the process is followed by the rest of image processing techniques until the result is obtain. For this programming, OpenCV library is used as the based to write the image processing with Python as the programming language. The camera used is Raspberry Pi camera and this camera has its Raspbian special writing language to call its function and manipulate its features such as brightness, contrast, image effect and add annotate-text. Ideally, a standardize OpenCV writing method used for the whole script is preferable. To convert the camera to OpenCV writing method, creating a new variable is needed to assign the name for the assignment. In this case, “camera = Picamera()” is used. This direct the programming that camera is the Picamera and it has Picamera’s array store within it. Then, use OpenCV writing method to write the rest of the capturing progress script. As reminder, Picamera also has its own libraries. Table 3.7 shows the library for Picamera.

Table 3.7: Library for Picamera

Number	Import library for Picamera
1	from picamera.array import PiRGBArray
2	From picamera import Picamera

3.6.2 Camera setup

Loading this programming will cause camera setup to be initiated for every single started running time. The main purpose of having camera setup procedure before image acquisition is to recheck the position of the camera. Camera calibration can be different in terms of angle and position every time after setup. To adjust the position to improve the quality of image that shows the entire hand for easy measurement, camera testing need to be included. Besides, camera setup can become a powerful tool helps to check the variety environmental condition such as room lighting. The lighting effects can obviously being check by viewing the output of camera setup, which refer to the testing image. Hence, the condition of lighting can be adjusted to maximize the reliability of measurement algorithm and reduce the error which halt the programming. Camera setup is a loop for unlimited adjustment on the camera calibration and environmental condition until the most acceptable image position and quality can be captured. Hence, it should have user input function to direct the programming whether it should continue looping within the camera setup or proceed to the next function.

3.6.3 Image acquisition

Image acquisition is to capture the user's hand by giving order to the camera. It means that image capturing of subject's hand is in real time. Under normal condition, the camera calibration and environmental condition have been revised in camera setup step. Hence, the only variable which can affect the quality of the image is the subject's hand. The person should be calm and position his/her hand with fingers open widely on top of the black background and be still. His/her fingers should never exceed the perimeter of the black board because it can cause the programming operation to wrongly detect the target from the image. The input for the image is only the hand without other regions such as watch, bracelet or cloth sleeve. These regions must to be avoided capturing into the image because it results to the target detect confusion during label components step.

3.6.4 Image pre-processing

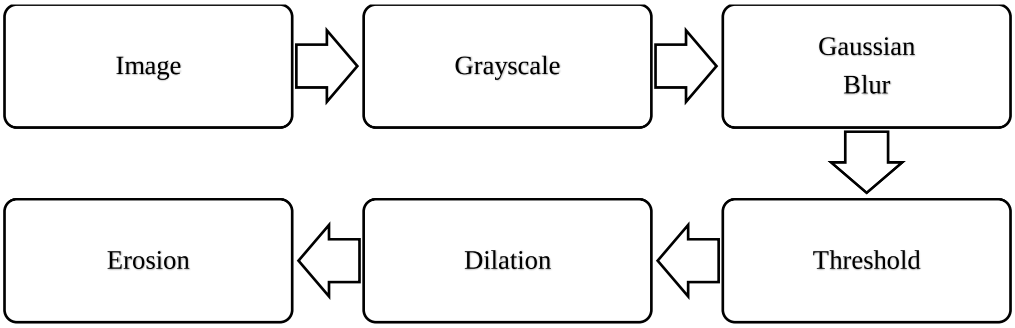


Figure 3.13: Pre-image processing

Figure 3.13 shows the flow for pre-image processing. Image pre-processing aims to improve the quality of the image by reducing the unwanted noise. The noise can be shadow, black dot, blur and even light contrast. The images sequence should be pre-processing in order to further improve the quality of images and extract useful information to gain better accuracy on the detection (Min & Principle, 2015).

After image acquisition, the image is converted to grayscale from three colors, BGR (Blue Green Red) sequence. The objective here is to kick out the fourth parameter, alpha(α) to reduce the burden of the median filtering. Besides, converting to grayscale also lower down the image pixels to ease the process of unwanted noise elimination during median filtering. The sequence of pre-processing is followed by application of Gaussian Blur. Gaussian Blur is a tool to do smoothing to filter background noise. It applies low-pass filters(LPF) to remove noise and blur the image and high-pass filters(HPF) in finding edges in the images. Hence, it will automatic maintaining the clarity of the image to avoid too smooth. The sequence is followed by thresholding. It is the well-known simplest segmentation method to separate the regions of an image corresponding to target analyzed objects. For this pre-processing, threshold binary is used to separate the hand and background by performing a comparison of variation of intensity between object pixels and background pixels. Figure 3.14 shows an example of thresholding.

Figure 3.14: Thresholding
(https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html)

The last two steps involve in the image pre-processing are the operations of morphological transformation. Morphological transformation is an image shape based operation that perform on binary image. It requires two inputs. They are the image for operation and structuring element or kernel to decide the nature of this operation. Basically, the morphologic technique used here is the variant forms, known as closing. It is the operation of Dilation followed by Erosion.

Dilation only focus on foreground to further remove the noise within the object. This operation will increase the size of the foreground object and the noise within the foreground object is vanished. Then, apply erosion. Erosion is just the opposite of dilation because it focuses on the background. It will erode the thickness boundary of the foreground object away after remove the noise on background. Normally the foreground object is white and background is black. In short, the combination of Dilation followed by Erosion can successfully remove the noise or black dots inside the foreground/white object while maintaining the size of the object. Figure 3.15 shows an example of closing morphological operation.

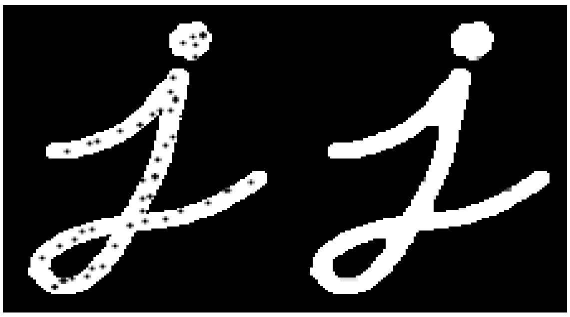


Figure 3.15: Closing morphological operation
(https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html)

3.6.5 Image classification

Image classification is to distinguish the main-focus foreground object from background. For this project, apply image classification is to detect hand from the image. A technique used is known as contour. To define, a contour is a curve joining all the continuous points, usually is along the boundary that is having the same color. Same color object gives same intensity. Contour is chosen due to its precise shape analysis, object detection and recognition. For example, contour is widely used for hand gesture recognition and object detection. To perform contour operation, usually binary image can output a better result with higher accuracy. It requires three arguments when calling this operation. The first is the source image, second is the contour retrieval mode and the third argument is index of contour. Third argument plays the main role to denote the contour approximation method, either all points on the line or just the end points on the line. Figure 3.16 shows the type of contour approximation method.

Figure 3.16: Type of contour approximation method
(https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html)

Drawing contour is easy and simple. It can be done by just calling the operation cv2.findContours. Given that the condition of hand is a non-regular shape, it is needed to decide if the hand on the image occupy the minimum or maximum area. Then, to display the contour found on the image, contour area is applied and it will detect area, whether it is minimum or maximum stated. To ensure the contour found is a closed shape curve, a second contour feature, contour perimeter is applied and it is followed by contour approximation. Contour approximation can be adjusted to specify the number of vertices depending on the precision. This adjustment is done by select the correct output for the epsilon percentage. A smaller percentage represents more number of vertices. There are many other contour features and every single has its own function. They will be discussed in the next sequence, label components. Figure 3.17 shows the original image, image with 10% epsilon and image with 1% epsilon.

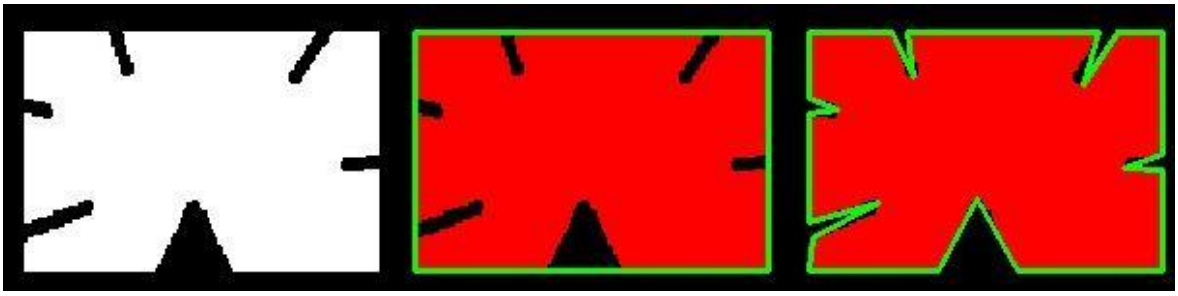


Figure 3.17: Original image, image with 10% epsilon and image with 1% epsilon (https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html)

3.6.6 Label components

Label components is to detect and label all the required points on the hand for measuring purpose. A hand is from the wrist extended through palm to the end of middle finger because it is the longest finger. To measure finger length, palm breadth and palm length, reference points are needed to direct and guide the programming to point out the structure tendons and the segmentation part of a hand. Hence, the first element point required is the center point of a hand. It can be identified using moment function of contour features. This image moments function provide solution to extract centroid of hand. This centroid is ideally found on the hand palm. Contours of hand must be extracted first in last step, image classification and pass the information to this function so that image moment will only focus on the hand to find its centroid point.

However, there is problem because the camera might capture more than just the hand but also part of arm. To minimize the error of centroid found, four extreme points on the hand will be determined and using these four extreme points with the help of contour of hand, the more accurate centroid of hand can be extracted. Four extreme points are north, south, east, west. Some developers refer it as top, bottom, left and right. To find four extreme points, the shape of hand is the key to decide which contour feature to apply. For hand, convex hull feature is applied.

Unlike contour approximation, convex hull checks a curve for convexity defects and corrects it. In short, the curve of convex hull is bulged out and this enable the fingertips to be identified. After applying convex hull, the function of four extreme points can be applied. This is because it is based on the convex hull to define the maximum four points in all four directions. Tuple function must be applied for Numpy array to read it. Afterwards, centroid can be found.

Getting only the centroid is far not enough to measure the three parameters and compute finger vs palm measurement. To track the point where is the middle finger located and how long it extended to, the finger valleys can be best solution to point out from the image. There is always finger valleys in between two fingers and it is easy to detect. By using the contour hull, it can use to track the finger valleys. Assume convex hull can detect all the fingertips, a reverse of its application can track the bulged in curves and indicate it with circles. All these circles are known a convexity defects.

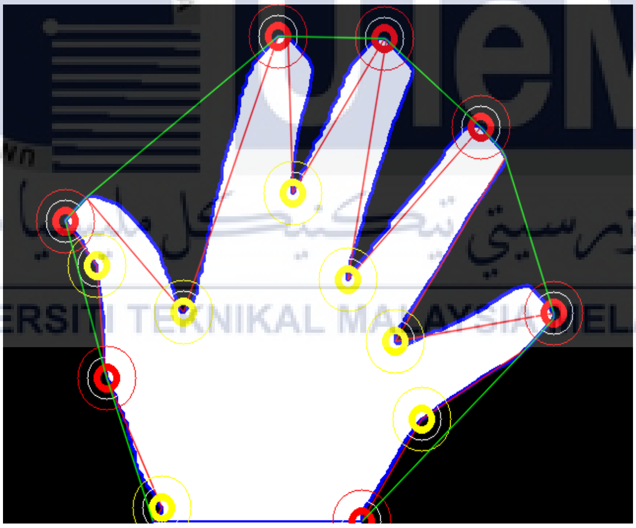


Figure 3.18: Results of convex hull and convexity application (https://www.researchgate.net/post/How_to_find_Convexity_defects)

Figure 3.18 shows the result of convex hull and convexity defect application. From the image, yellow circle shows the defects and red circle show the fingertips. To find convexity defects, the “returnPoints” on the hull syntax need to change to false. It is followed by declare the defects function and using a loop to track start, end, far and approximate distance to the farthest points to obtain defect shape. All the defects in between the finger valleys are defect far points.

3.6.7 Measurement

Measurement is taken by using mathematics method to get the distance of three required parameters based on the hand image. The parameters are finger length, palm breadth and palm length. All the measurements are first measure in millimeter. Based on hand shape, middle fingertips always to be the extreme top. Having finger length and palm length can start to calculate for the last measurement, which is finger vs palm (shape) index. Figure 3.19 shows the parameters and the formula for finger vs palm (shape) index.

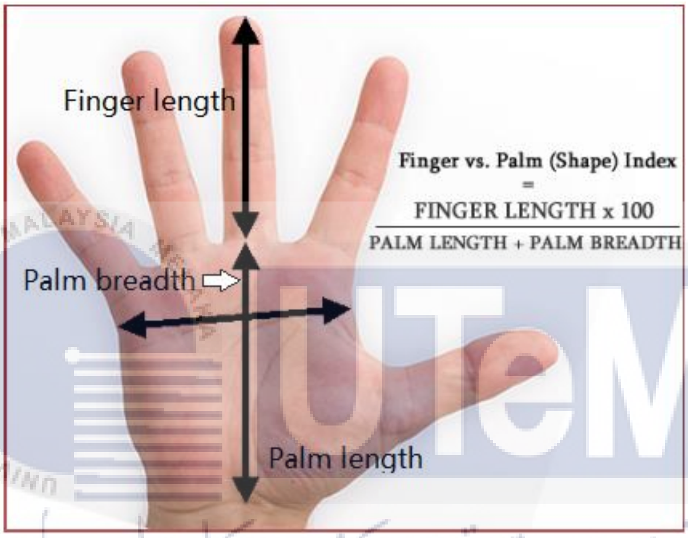


Figure 3.19: Parameters and the formula for finger vs palm (shape) index

Finger length is the distance from extreme top to the midpoint of two defect points on the hand. First, extract these two defect points and then calculate the midpoint of these two points. Getting the distance based on the coordinates of the midpoint to the extreme top. To get the distance, Pythagorean theorem is used. This is because both points are array form. The distance found is finger length.

Palm length is the distance between the bottom point of finger length and the wrist. Thus, a rectangle of contour feature is drawn. This contour feature enables only the rectangle draw follow the size of the hand from the extreme top to the wrist. Get the height of the rectangle. Using basic python mathematics deduction to minus finger length from the height, palm length is measure. The result of this deduction is the palm length.

Palm breadth is difficult to get but luckily it is the shortest length on the x-axis on hand palm. Using this characteristic and create a loop to distinguish it. Draw a horizontal line from hull touches edge on palm until it touches the minimum hull line at the end of the other side of hand. The line forms and its length is equal to palm breadth.

Finger vs palm (shape) index can be calculated based on the finger length and hand breadth measurement. This index only need to apply with formula on the bottom of palm breadth. Then, compute the all found measurements to real world physical measurement by apply pixels per metric. Lastly, convert the unit to millimeter. Every single measurement will have an additional time taken executive function before the measurement starts and stops when the parameter has obtained.

3.6.8 Output result

Output result is to display the hand measurement and time taken from the source code to the user. This is to provide convenience to the user to get the result in a faster and easier way. Therefore, print function is added for every measurement and the result is display on the terminal. Then, the results are all corrected to 2 decimals places using round method to avoid confusing and save time when transfer the data for data analysis.

3.6.9 End

Obtaining the measurement and time taken also mark the full stop for the sequence of image processing technique in programming development. Hence, after this programming development is complete, it is followed by the next flow, system testing based on the project planning.

3.7 SYSTEM TESTING

This system consists of hardware and software categories. Therefore, the hardware part must be well setup and connect to the laptop to link the software part and display on laptop. On this section, testing method will start from experimental setup, camera testing, experimental procedure and lastly expected result. The experimental setup will explain how to setup the hardware and software part to run as a system. Then, checking the functionality of camera to ensure it is working as expected. During the experimental procedure, programming is under checking and testing. If the output of this system is running as expected, the result in the expected range and no surprise or error pops up, system testing is given a pass. If these three requirements cannot be fulfilled, system testing is a failure and overall checking takes place. The checking will start from system development to diagnose the root of the problem, whether it is coming from hardware or software environment. Any further correction or modification will be carried out and redo system testing until the system is stable, which also means it is working as expected.

3.7.1 Experimental setup

To capture a still image of hand, it is best to have a place to position and rest the hand of a subject. Since image processing works best with big intensity difference, the background of the image should be dark color, for example black background. This brings a big contrast of skin color with the black background and certainly this should work best if it is under the condition of sufficient light. To fulfill the conditions, a standard A4 size black board is used as the place to put user’s hand. Logically, the black board is an indicator for user to know the black board is the place to position hand and the fingers should within the size of the black board. A bigger size of white paper is put behind the black board to cover space which is bigger than the black board but captured in the image. Since the hand is on resting condition, the camera will be placed on a higher position to capture the full hand image. After several times of testing with the image capture, the best height and length has decided. The height is 34cm from the black board. Length is 11.5cm measured from the top edge of black board to the center point of the camera. Figure 3.20 shows the experimental setup and figure 3.21 shows the height and length of the hardware setup.

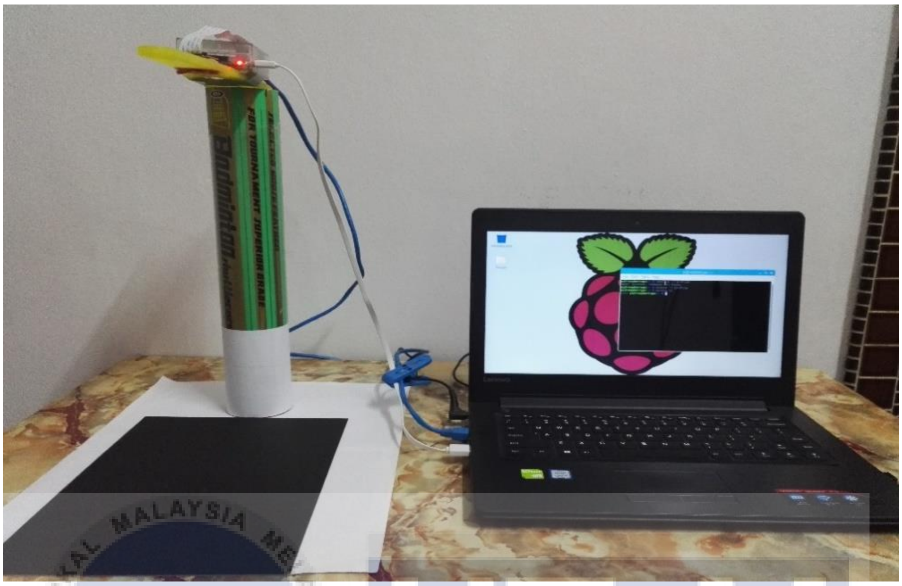


Figure 3.20: Experimental setup

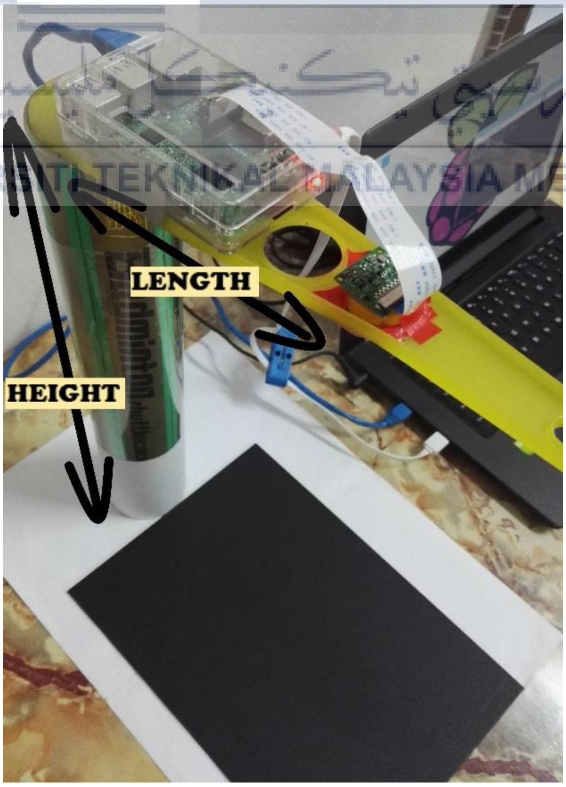


Figure 3.21: Height and length of the hardware setup

3.7.2 Camera testing

Completing experimental setup continues with camera testing. Camera testing is crucial step to avoid misleading or misunderstanding when camera failure occurs during testing programming. Camera is linked to the Raspberry Pi controller and to test it need to remote access through laptop. The procedure for image capture is shown below.

1. Open the terminal in Raspberry Pi screen
2. Type this command, “raspistill -o image.jpg”
3. Wait for the command to run.
4. Go to the home/pi to look for the image named image.jpg.
5. Open and view the image to check for the quality of the image resolution.

After camera functionality checking is done, it is important to check if the image captured is at the right angle and position. Adjustment on camera calibration can correct the angle and position to gain a good image. Figure 3.22 shows the correct angle and position of image captured.

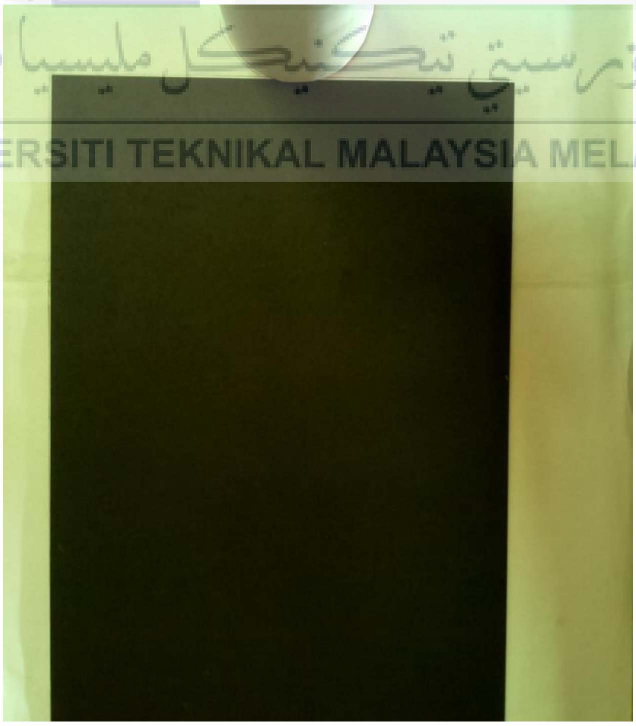


Figure 3.22: Correct angle and position of image captured

3.7.3 Experimental procedure

Experimental procedure shows the steps guiding the user how to use this system to get hand measurement. Basically, it involves only four main steps while the first and last step are for preparation of setup purpose and conclusion of experimental procedure respectively. Figure 3.23 shows the experimental procedure.

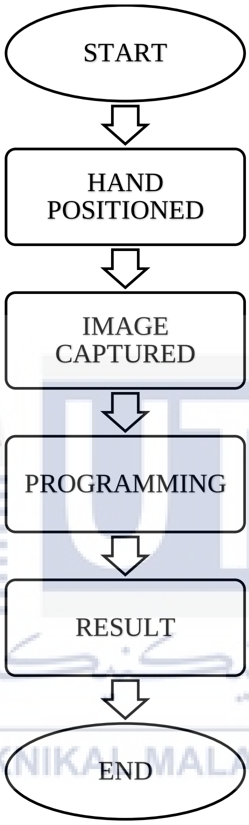


Figure 3.23: Experimental procedure

3.7.3.1 Start

The beginning of experimental procedure depicts the well setup of system and it is ready for use. Well setup of system includes the camera calibration, Raspberry Pi screen remote access and the calling of source code to set the system in ready mode to accept input.

3.7.3.2 Hand positioned

Hand positioned means the user's hand is resting on the black board with the correct orientation within the range of the image captured area. User's hand must be placed with hand back facing the ground and hand front facing the camera. This is to minimize the wrong reading due to the sharp nails on the fingertips. Besides, users should try to spread all five fingers while placing it on the black board. Spreading fingers widely drives to getting a better finger tracking because of the easiness of convexity defects to locate bulged in position and define it as a defect point. Avoid capture the non-hand-type on the wrist image. These things maybe a watch, bracelet, thread, finger ring, wrapped or even cloth sleeve. If possible, remove it or else try to avoid captured in the image.

3.7.3.3 Image captured

Preferably, user's hand is still without shaking and place calmly with resting condition on top of the black board. The hand with spreading fingers must be within the boarder of the black board. An image is captured after an order has been made to run the source code. Just wait for 5 seconds until an image showing hand is display on remote access Raspberry Pi screen.

3.7.3.4 Programming

Image processing technique in the source code is initiated when the hand image captured and enter is pressed. The image will run through loops and filter to remove noise and finally measurement will take place. Figure 3.24 shows progress of experimental procedure.

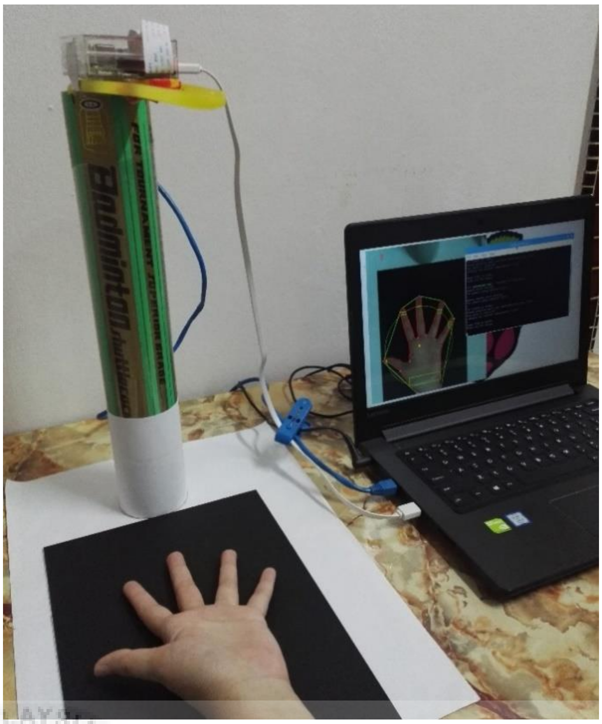


Figure 3.24: Progress of experimental procedure

3.7.3.5 Result

Result is displayed on the terminal if no error pops up. The result shown includes all three hand measurements and the time taken. The hand measurements shown are start from finger length with its time taken, palm length with its time taken and lastly palm breadth with its time taken. Then, finger vs. palm(shape) index is calculated and displayed on terminal. If the result displayed is less than three categories and without finger vs. palm(shape) index, a bug occurred and a deeper rechecking will from system development and programming will be redo.

3.7.3.6 End

The program will pause and wait for an input. This input inserted gives continue or terminate order to this program. Check for saved image on the project file on home/pi/desktop/project.

3.7.4 Expected result

Expected result is achieve when the result of experimental procedure, in terms of time and hand parameter measurement are displayed according to finger length, palm length, palm breadth and finger vs palm(shape) index. User can compare the online result with manual estimation using ruler or tape or Vernier caliper to verify the measurement. For the time taken, perhaps he can use his/her stopwatch application installed on phone while taking measuring data. The time taken using system is expected to take less time compare to manual measurement. For hand measurement, the decimal places displayed on terminal for sure is more than the decimal places can be measurement using Vernier caliper, which has maximum 2 decimal places. In fact, the data for vision measurement will be rounding to 2 decimal places except for the finger vs palm (shape) index, it has up to four decimal places.

3.8 DATA ANALYSIS AND VERIFICATION

Data analysis and verification is using the results getting from manual method and vision method to do comparison. A total of 30 subjects will be randomly selected to measure their right hand. All of them must be Asian people. For vision measuring, they need to position their hand and program the image follows the simple steps discussed in the experimental procedure in getting the results. For the manual measuring, a Vernier caliper will be used to ensure their hand. Two persons in a group with one person controls the time taken and the other person does the three hand parameter measurements. Once the measurement is begun, time taken is recorded and stop when getting the hand measurement. Then, finger vs palm(shape) index is calculated manually one by one. Every subject will be listed a number as an indication to tabulate the data collected for both manual and vision method and to compare their time frame and hand parameters for analysis.

Both results for the same subject will be used to compare and do analysis. Description and graph are used to explain and provide a better understanding for the analysis. Time taken for both data will be tabulated to do compare and verification. Mean will be calculated to do time taken comparison. Bland-Altman plot is used to analyze both of the two of manual and vision method whether they have an agreement between two quantitative limits.

3.9 SUMMARY

To conclude, methodology is about the complete work flow for the image processing technique. It has 6 main stages. They are starting from literature review, project planning, system development, programming development, system testing and lastly data analysis and verification. Every stage is discussed in detail to reveal the idea and importance of it. All these stages are important because an error in a stage can lead to the problem in the final stage. Hence, patient and precautions need when carry out the experiment based on methodology in getting results. Lastly, methodology is the guideline to set up the experiment for collecting data.

CHAPTER 4

RESULT AND DISCUSSION

4.1 INTRODUCTION

This chapter marks the start of the second part of the project. It shows the output of this project and discusses it using graphs and descriptions. The output is generated using the sequence implemented in chapter 3. Then, it will be discussed based on the objectives of this project. The structure used to write this chapter has 3 main categories. They are programming, data collection, comparison and analysis on data collection. The 1st objective is involved in programming sub-chapter while 2nd and 3rd objectives are included in comparison and analysis on data collected. Data collection consists of vision reading and manual reading. It is a sub-chapter justifies reliability of vision measurement and manual measurement. It also shows the results. Photos of the taking manual reading are attached. Comparison and analysis on data collection will be discussed.

4.2 PROGRAMMING

Programming is the core of software and the chances of pass depends on it. To develop a systematic programming, designing the flow is crucial. This flow includes image capture, image processing technique and outsource the result. As explained in chapter 3, the sequence of image processing technique is used to process the image and output result. Hence, the flow on how to direct the hardware to capture an image, where to store image and how to link image processing technique in this sequence and display the result need to consider and map out a complete and systematic flowchart. A complete source code is attached in Appendix E. Figure 4.1 shows the flowchart of the programming.

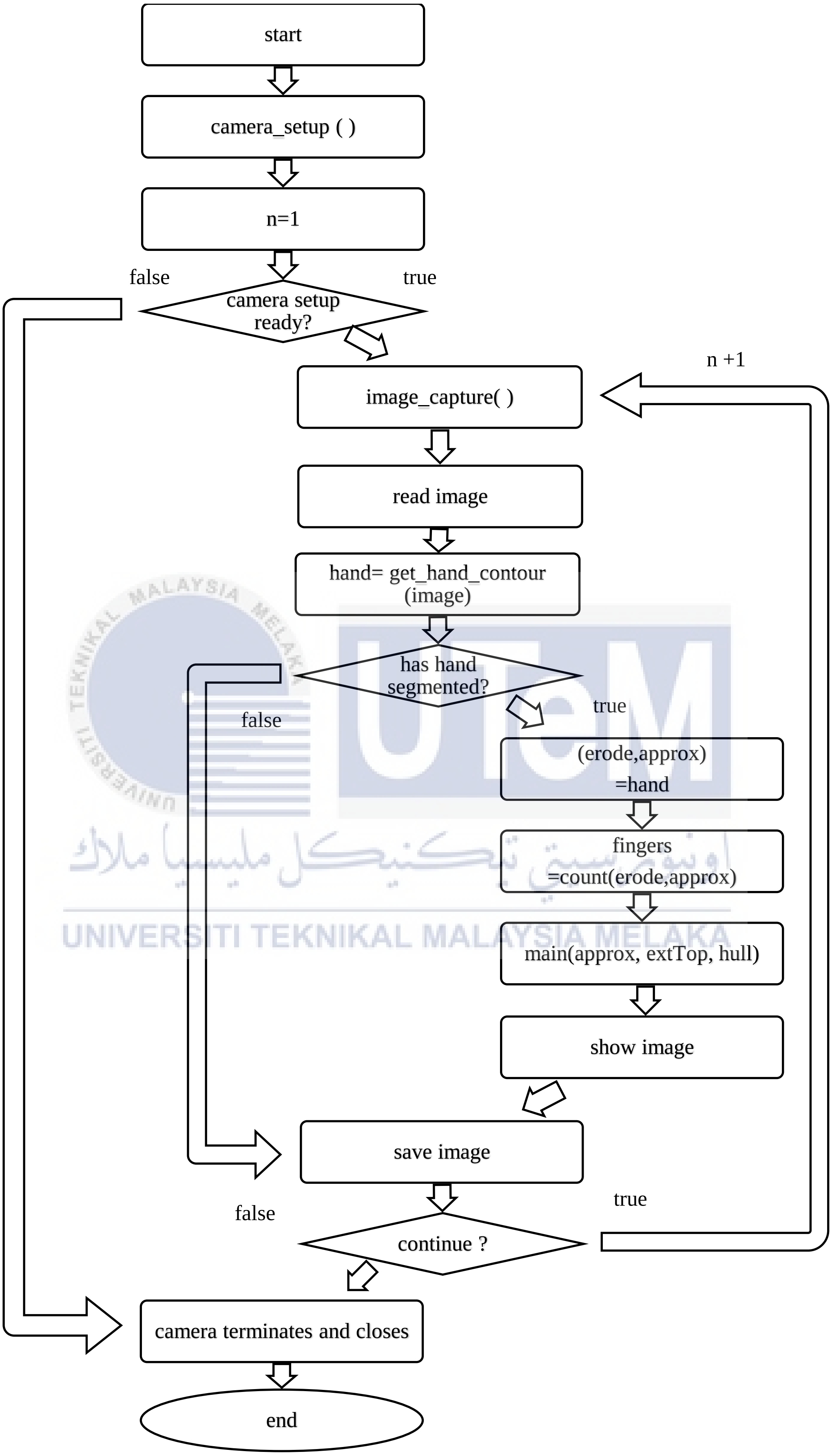


Figure 4.1: Flowchart of the programming

4.2.1 Flowchart explanation

In this section, every single step shown in working procedure flowchart will be explain. According to Python usage standard, the main function use to compute the sub functions is known as "if__name__== "__main__" . Define function (def) is applied for sub function that requires loop to solve or involves long sequence in getting the output. For example, camera_setup, image_capture ( ), hand=get_hand_contour (image), fingers =count (erode, approx) and main (approx., extTop, hull). Since image captured is done in a define function, the saved image cannot be read and displayed in the main function or another sub function. Therefore, path function is applied. Path function constructs an absolute path to the directory of the file located to save the captured image, read it and access it for measurement. The overall programming is tested and applied in Virtualenz (CV) environment.

4.2.1.1 start

First, open the Raspberry Pi terminal and enter Virtualenz(CV) environment. It can be done by applied command "source ~/. profile" followed by command "workon cv". Then, calling the source code by applied command "python pi_testing.py". The command "pi_testing" is name of the source code for this programming. This name of source code can be changed and a changing of it should edit on the command to call this programming.

4.2.1.2 camera_setup

This function is initialized at the beginning of the programming to double check the camera calibration. Define function is applied for this function, "camera_setup". A loop is involved to recapture the image until correct orientation and position is attained. Example of the image can refer to chapter 3 with the sub-sub title 3.7.2 camera testing.

4.2.1.3 n =1

To declare n is equal to 1. This variable is used to save the image in a sequence starting from number 1 to prevent the program from rewrite the previous saved image.

4.2.1.4 decision making 1: camera setup ready?

This decision making is to ensure the hardware and software coordination are stable and achieve before proceeding. If there is a bug blocks or disturbs the function, this program will terminate the camera and quit the program.

4.2.1.5 image_capture ( )

Define function is used. This function is for hand capturing purpose. It doesn't involve any loop for repetition. The image captured is store to the path and temporary name it as "Subject" in png format. The color is convert from picamera format, RGB to OpenCV format, BGR. Figure 4.2 shows the image captured.

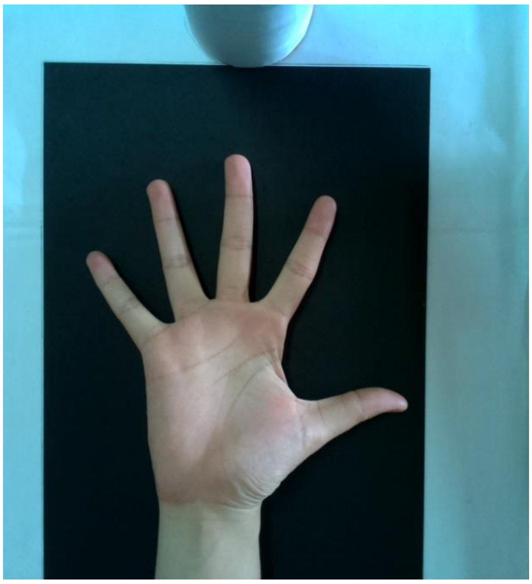


Figure 4.2: Image captured

4.2.1.6 read image

In the main function, image is declare as the variable to read the image captured in image_capture ( ) function. This is done by direct image to the path where the image is stored.

4.2.1.7 hand=get_hand_contour (image)

Define function is applied here. “hand” is the variable used to call the sub function get_hand_contour (image). In the sub function, image captured is read using the path. Afterwards, a sequence of image processing sequence occurs with aim to remove the noise and draw contour. Eventually, this sub function “get_hand_contour(image)” will return the threshold and approx information to the hand. The sequence is start with grayscale, GaussianBlur, threshold, dilation, erosion and finally draw a contour. Figure 4.3 shows the flow in get_hand_contour(image) sub function.

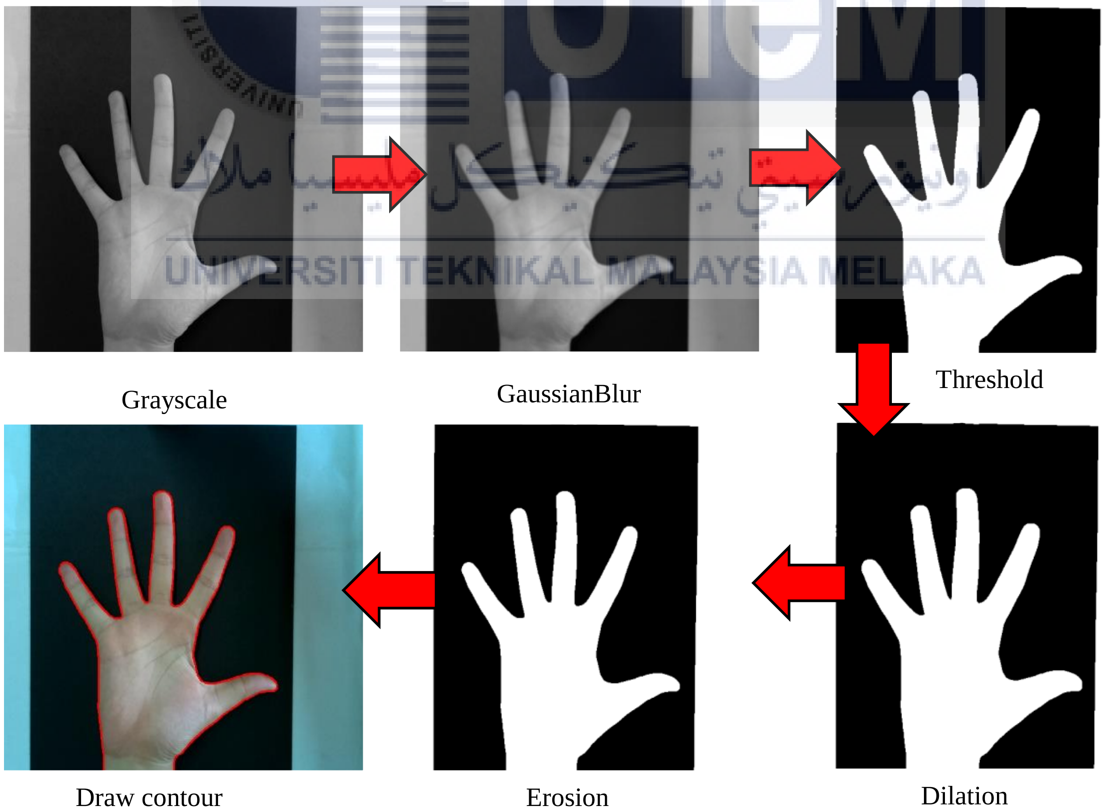


Figure 4.3: Flow in get_hand_contour(image) sub function.

4.2.1.8 decision making 2: has hand segmented?

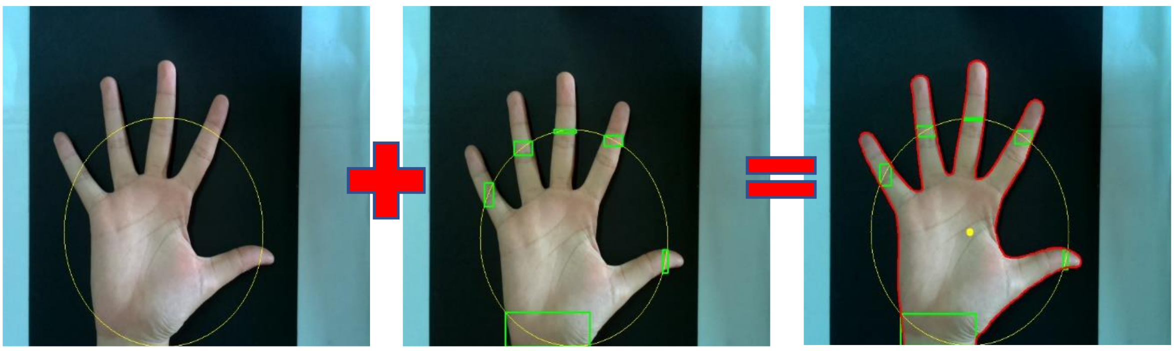
This is the 2nd decision making. "If" loop is used to check the hand segmentation in the image. If the hand has not been segmented, the following sub functions skip and directly carry out save image function. Under normal circumstance, the segmentation should have been done and the next few functions will be executed. Each individual function in this loop has their own tasks and this loop help to ensure tasks are completed.

4.2.1.9 (erode, approx) = hand

The (erode and approx) is the output from the get_hand_contour (image) function. The previous function make sure only segmented hand can pass through and this function is to unpack erode and segmented region to hand.

4.2.1.10 fingers = count (erode, approx)

Define function is used for this function. The main purpose of this step is to verify the object detected is a hand. This technique is using the combination of drawing a big circle using the centroid as the center point and drawing rectangle when it encounters contour along the circumference of big circle. The number of rectangle is the number of finger count with the condition that rectangle found on the wrist is not counted. Figure 4.4 shows the combination of drawing circle and rectangle to count the number of fingers.



Drawing a big circle | Drawing rectangle | Counting rectangle

Figure 4.4: Combination of drawing circle and rectangle to count the number of fingers

Generally, it starts with finding the centroid on hand. Four extreme points from contour feature are used to detect the four directions. Centroid is found by finding the midpoint along x-axis and midpoint along y-axis. Draw the centroid.

Then, to draw a big circle, the radius must be identified. The centroid will be used as center point. Euclidean distance formula is used to calculate the maximum distance between centroid and the four extreme points. Assumption of average Asian finger length is within 60% of the maximum distance between centroid and four extreme points. Therefore, radius is equal to 60% of the maximum distance between centroid and four extreme points.

Calculate for circumference to draw the rectangle along the circle line. Basically, circumference is calculated based on the radius with a formula, $2\pi r$. Draw the big circle using the radius found.

Finally, compute rectangle along the circumference when it detects contour region of hand. If the number of points along the contour exceed 25% of the circumference, it is defined as the rectangle along the wrist. This rectangle will not be counted. Hence, every rectangle drawn is counted except the rectangle along the wrist.

This verification is using fingers tracking method as the key to validate the object is a hand. Hand has five fingers and hence if five rectangles are counted, the object is verified as hand. However, if less than 5 rectangles detect, the program will terminate with a statement "Error! This is not a hand with 5 fingers!" If hand is detected, 2 parameters will return for next function usage. Figure 4.5 depicts the flowchart for hand verification. The illustration of the image may give a better understanding on how to use image processing technique to count fingers for verification. Figure 4.6 shows image processing technique on hand verification.

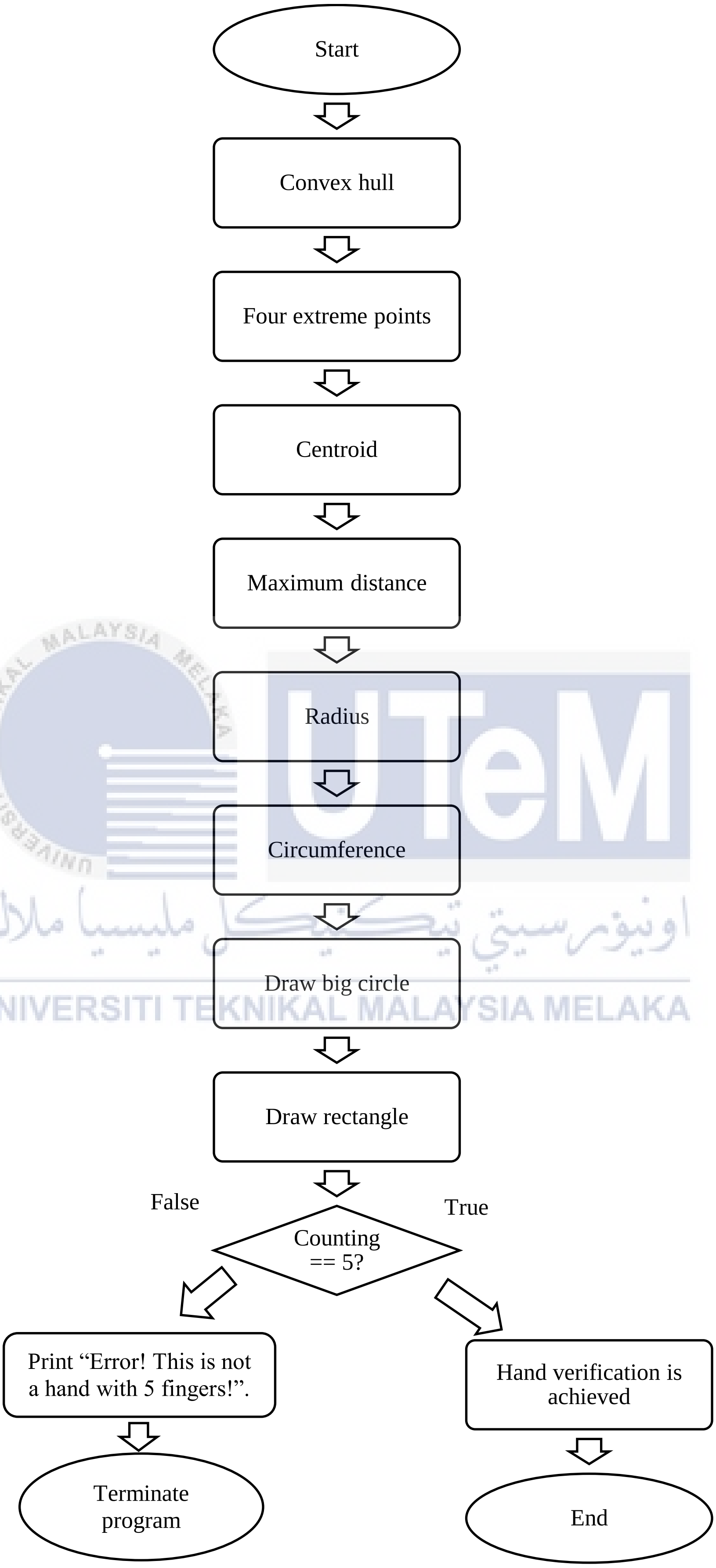


Figure 4.5: Flowchart for hand verification

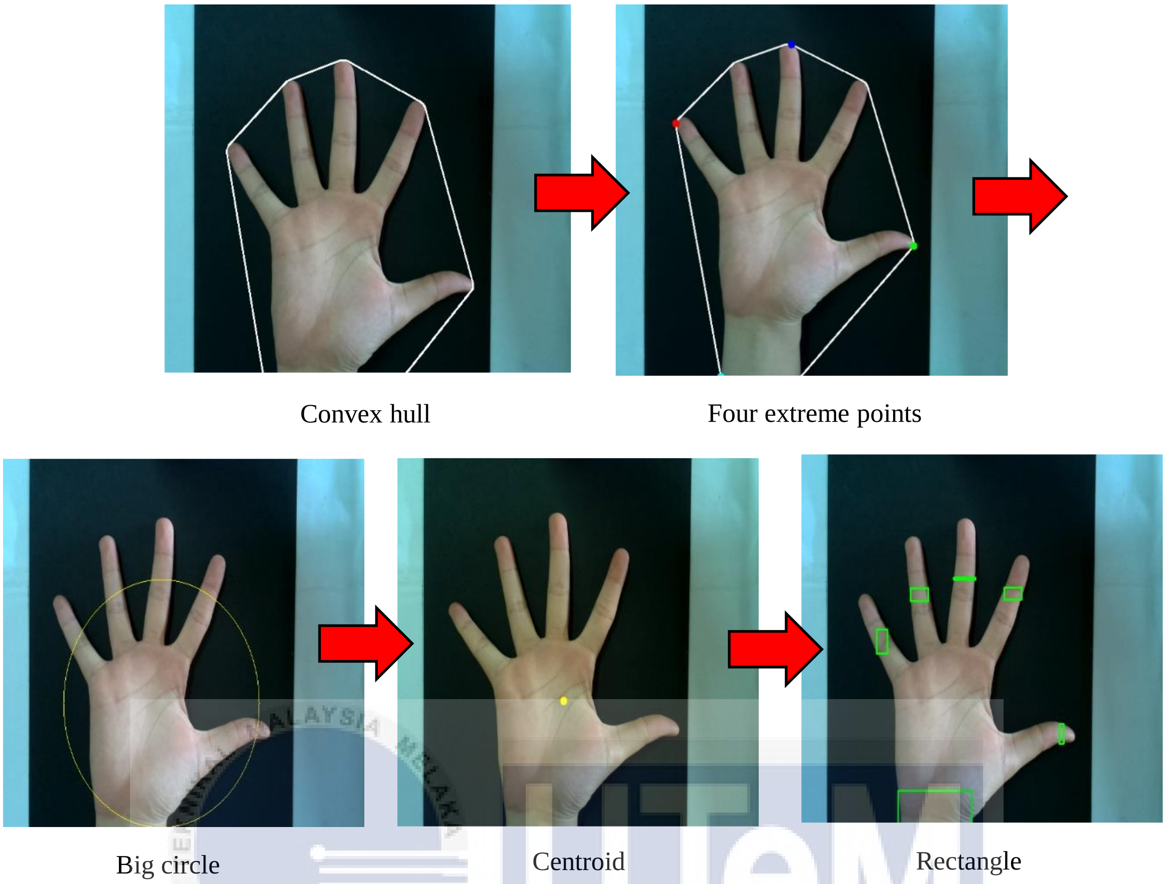


Figure 4.6: Image processing technique on hand verification

4.2.1.11 main (approx, extTop, hull)

This is the last function using define function. It is used to calculate the measurement of finger length, palm breadth, palm length and finger vs. palm (shape) index. It inputs (approx, extTop and hull) from the previous function to do calculation. First, it defines all the required variables such as time and pixels per metric. Time function is needed to record time taken for every measurement taken. The hand parameters are measured in pixel and converted from pixels to metric using pixels per metric function.

For finger length, hull and defects points need to clarify to locate the points on hand in the image. Figure 4.7 shows the flowchart on how to find far defect points. From figure 4.6, the purple dots are far defect points and white line is hull convex line.

Draw hull Draw far defect Combination

Figure 4.7: Flowchart on how to find far defect points

Having far defect points, the distance between the midpoint of defect points located at the edge of middle finger and the extreme top can be measured. This distance is the finger length. Then, using rectangular shape to map the on the hand where the top is the middle fingertip and bottom is the wrist line. Get the height of the rectangle. Palm length is the distance using height of rectangle to deduct the finger length. Finally, palm width is using the shortest length of the hand contour on the x-axis and located on the palm area. Finger vs palm (shape) index is using the formula (finger length *100) / (palm length + hand breadth).

Time taken function is used for every measurement and the measurements are converted to millimeter before using print function to display them.

4.2.1.12 show image

This function is written to display the output of the image after using image processing to draw all the points. Figure 4.8 shows an example of the complete image.

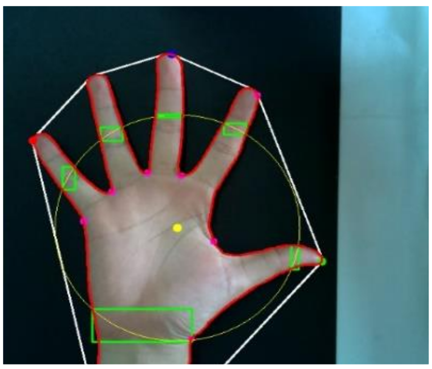


Figure 4.8: An example of the complete image

4.2.1.13 save image

This step is to save the hand image to the same path. It means that the previous taken image will be replaced by this new image. Using os library can rename the previous image and replace it with the new image.

4.2.1.14 decision making 3: continue?

This is the last decision making. This program will print "Continue for another measurement?" and wait for an input. If the input breaks this loop, it will quit and proceed to the next function. However, if this loop continues, it will loop back to the image_capture( ) function and proceed the programming.

4.2.1.15 camera terminates and closes

This step is to terminate and close the camera. The camera can only be initiate again when restart this whole program starting from camera_setup.

4.2.1.16 end

The program will close and the terminal on Raspberrry Pi is ready to accept another new command.

4.3 DATA COLLECTION

Data collection has two categories. They are vision reading and manual reading. Obviously, vision reading was collected using programming application and manual reading was recorded using Vernier caliper to measure. Both data were taken from thirty subjects at the same time in the same place. Table 4.1 shows the vision measurement data collection.

Table 4.1: Vision measurement data collection

Subject	Finger Length		Palm Length		Palm Breadth		Finger vs Palm (Shape) Index	
	Time(s)	Measure. (mm)	Time(s)	Measure. (mm)	Time(s)	Measure. (mm)	Time(s)	(FL*100)/(PL+PB)
1	5.98	81.68	2.00	111.08	5.10	93.01	0.80	40.0216
2	6.00	67.00	1.98	88.50	5.40	73.50	1.03	41.3580
3	5.71	79.03	2.42	99.03	5.55	71.00	1.09	46.4800
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
5	5.99	81.03	2.30	98.86	5.99	77.30	0.78	45.9980
6	6.10	68.88	2.43	89.00	5.32	66.20	0.99	44.3814
7	6.04	71.00	2.33	90.79	5.78	70.00	0.86	44.1570
8	5.69	81.09	1.99	105.66	5.99	91.23	0.96	41.1854
9	6.00	79.56	2.01	99.50	5.43	77.66	1.05	44.9086
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
12	6.11	70.91	2.28	95.60	5.47	70.00	0.76	42.8200
13	5.96	67.06	2.10	97.00	5.76	68.31	1.06	40.5662
14	6.09	71.33	2.09	105.60	5.55	71.11	1.11	40.3656
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
16	5.81	75.17	2.11	91.00	5.97	73.50	1.08	45.6960
17	5.93	66.37	2.06	93.30	6.00	72.90	0.99	39.9338
18	6.14	74.25	2.21	99.59	6.02	76.00	0.78	42.2860
19	6.15	69.50	2.24	94.90	5.46	74.38	0.80	41.0562
20	5.92	70.90	2.31	91.66	5.83	70.40	1.00	43.7492
21	6.00	72.09	2.36	102.05	5.88	68.00	0.76	42.3934
22	5.75	72.10	2.08	91.42	5.35	73.55	0.78	43.7049
23	5.90	70.00	2.15	94.00	5.38	74.99	1.02	41.4226
24	6.09	75.50	2.19	99.67	5.44	73.01	1.06	43.7225
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
26	6.14	82.06	2.40	112.35	5.76	87.88	0.86	40.9829
27	6.06	75.07	2.22	93.00	5.48	78.10	0.89	43.8749
28	5.79	87.60	2.19	109.25	5.51	85.00	0.93	45.0965
29	5.87	82.00	2.05	118.03	6.00	90.36	0.92	39.3493
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

For vision measurement, time and the three main measurements, which are finger length, palm length and palm breadth were calculated automatically. Finger vs palm (shape) index was the last to generate because it required the output of three main measurements. To standardized the precision of vision measurement, rounding decimal places was applied. All the measurements have been tabulated and rounded up the decimal places to only 2 decimals places except for finger vs palm (shape) index, which has four decimal places. In table 4.1, the data label with red color have error and thus the measurement cannot be generated from the image. Table 4.2 shows the manual measurement data collection.

Table 4.2: Manual measurement data collection

Subject	Finger Length		Palm Length		Palm Breadth		Finger vs Palm (Shape) Index	
	Time(s)	Measure. (mm)	Time(s)	Measure. (mm)	Time(s)	Measure. (mm)	Time(s)	(FL*100)/(PL+PB)
1	10.10	82.00	25.70	112.00	6.70	95.00	10.50	39.6135
2	14.00	67.50	15.10	88.50	7.50	75.00	9.74	41.2844
3	10.10	79.00	8.50	99.00	8.60	71.50	9.00	46.3343
4	11.70	75.00	10.80	91.00	10.30	76.50	8.70	44.7761
5	10.00	80.00	8.80	99.00	7.40	79.00	10.34	44.9438
6	13.20	69.00	12.20	89.00	10.60	67.50	11.00	44.0895
7	10.70	71.50	8.10	91.00	8.80	69.50	10.80	44.5483
8	11.50	80.00	7.60	97.00	9.20	72.00	9.90	47.3373
9	16.20	79.50	5.70	99.50	10.10	78.00	8.98	44.7887
10	10.90	84.00	9.70	106.50	9.10	94.00	11.03	41.8953
11	11.80	70.10	16.60	99.00	10.60	75.10	10.22	40.2642
12	16.48	71.60	10.37	95.80	9.37	70.60	9.78	43.0288
13	15.90	67.00	9.52	97.30	10.02	69.00	10.01	40.2886
14	12.74	71.30	9.60	105.60	11.69	71.20	8.88	40.3281
15	10.20	75.70	10.80	90.60	10.80	70.00	9.00	47.1357
16	7.80	75.10	12.30	91.40	10.30	74.50	11.40	45.2682
17	10.20	66.50	8.90	93.50	8.80	74.90	8.90	39.4893
18	8.10	74.50	9.30	99.70	9.60	76.60	9.50	42.2575
19	8.20	69.50	13.50	94.90	7.40	74.30	9.76	41.0757
20	10.30	70.90	12.00	91.60	10.20	70.40	9.21	43.7654
21	11.10	72.00	12.70	102.00	9.70	68.00	10.44	42.3529
22	9.30	73.00	8.00	91.50	6.80	74.00	11.09	44.1088
23	14.00	70.30	19.20	94.20	14.50	75.15	11.18	41.5117
24	10.20	76.50	14.10	99.80	8.40	74.20	9.09	43.9655
25	10.20	78.00	7.50	106.10	7.00	88.00	9.13	40.1855
26	12.20	82.00	15.10	112.40	6.50	88.30	9.08	40.8570
27	15.30	75.00	7.80	93.00	9.30	78.40	9.65	43.7573
28	12.80	87.60	8.20	109.30	6.10	85.00	10.66	45.0849
29	9.20	82.00	9.90	118.00	5.40	91.00	10.20	39.2344
30	8.80	72.00	6.70	91.90	5.50	75.00	10.45	43.1396

Based on the data collected using manual measurement, the time was taken using stopwatch on phone application. This method is excellent but it requires a helper to press start and terminate the time recording because a stopwatch is not automatic. The maximum precision of manual measurement is two decimal places whereby the last digit is very difficult to read. For finger vs palm (shape) index, the measurement is calculate using a scientific calculator and the decimal places is round up to four decimal places.

4.3.1 Vision reading

Vision reading is the data collected from 30 subjects in UTeM library using vision measurement. Due to the time constraint to finish this project before due date, the vision measurement was carried out using offline method. Thus, only image captured using the same controller and camera was carried out in library. All the 30 images stored were process using programming and it has been run for 30th times. For the result, it consists of success and failure to process the images. Table 4.3 shows the result for vision reading.

Table 4.3: Result for vision reading

Subject	Success	Failure
1	X	
2	X	
3	X	
4		X
5	X	
6	X	
7	X	
8	X	
9	X	
10		X
11		X
12	X	
13	X	
14	X	
15		X
16	X	
17	X	
18	X	
19	X	
20	X	
21	X	
22	X	
23	X	
24	X	
25		X
26	X	
27	X	
28	X	
29	X	
30		X
Total	24	6

For 4th, 10th, 11th ,15th and 30th times, the contour detected the wrong spot instead of the hand. The program did not have any obstacle to progress. It showed no result but an error. This error stated that object does not have shape, hence no numpy array found. Assumption has been made for this failure. This may cause by the lighting or the shadow which drives the programming to detect wrong spot. Figure 4.9 shows the percentage for vision reading. This percentage consists of 80% success rate of blue component and 20% failure of orange component in the pie chart.

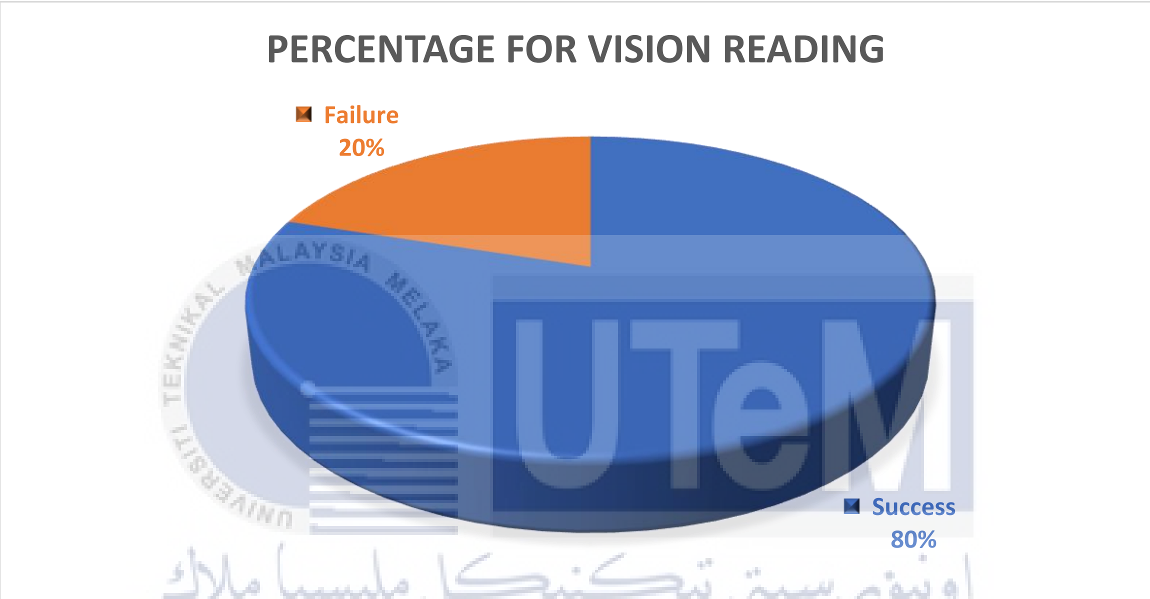


Figure 4.9: Percentage for vision reading

4.3.2 Manual reading

Generally, manual reading is the data collected from the same group of 30 subjects in UTeM library using manual measurement. This measurement was carried out directly after an image of hand had been capture. 2 persons need to work together to take manual measurement and time taken at the time. One person did the measurement and the other person controlled time variable and did recording. The result was recorded for further analysis and to calculate finger vs. palm (shape) index. Figures 4.10 shows the process of taking manual reading. Figure 4.11 shows the palm length measurement using Vernier caliper. Figure 4.12 shows data recorded and using phone timer to record time.

Figures 4.10: Process of taking manual reading

Figure 4.11: Palm length measurement using Vernier caliper

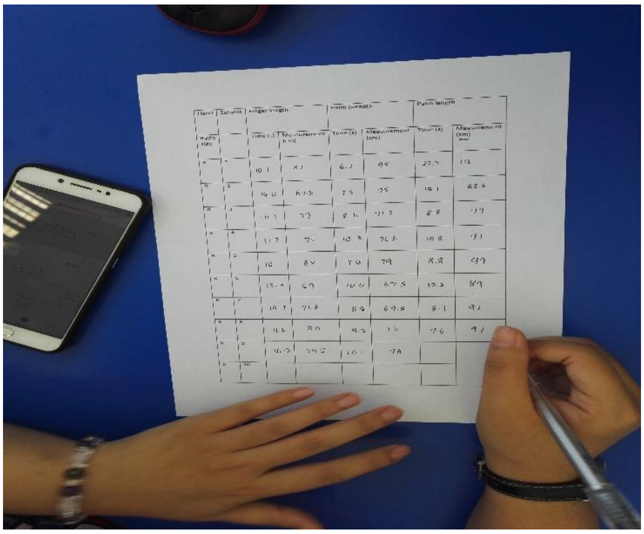


Figure 4.12: Data recorded and using phone timer to record time

Manual reading did not have any measurement failure and this mark a fully 100% successful rate in collecting data. Tabulated success and failure table will skip because it is not needed but a pie chart showing 100% successful rate will be included. Figure 4.13 shows the percentage for manual reading.

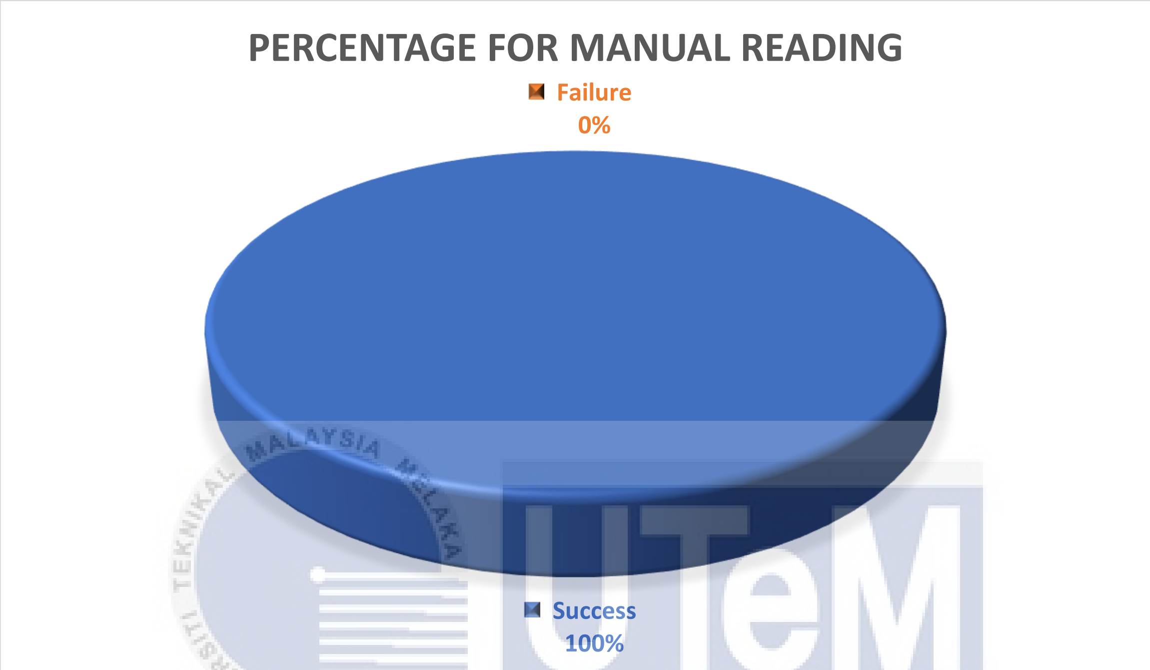


Figure 4.13: Percentage for manual reading

4.4 COMPARISON AND ANALYSIS ON DATA COLLECTION

Comparison and analysis on data collection is to identify a statistically significant difference between manual measuring and vision measuring method. This section will be divided into two parts, duration needed on taking measurement and analyzed and compare two type of hand measuring methods.

Mean approach is used to do analysis on duration needed on taking measurement. the results from both methods are compare and justify. For the second part, statistical comparison of these data requires a summarizing decision on how to observe the differences. Given the condition in this project, the subjects for two groups of data are same. The difference is the method used to get carried out measurement. Hence, Bland-Altman approach is applied. Data is analyzed using unit difference plot. This plot defines the intervals of agreements, which are lower and upper limit of agreements that comprises the intervals to evaluate the bias between the mean differences.

4.4.1 Duration needed on taking measurement

Duration needed on taking measurement can be analyzed by comparing manual measurement and vision measurement. 30 data have been tabulated and the time taken for every hand measurement will be summarized to perform mean calculation. Figure 4.14 shows the formula of mean.

$$\bar{x} = \frac{\sum x}{N}$$

$\sum x =$ the sum of x

$N =$ number of data

Figure 4.14: Formula of mean

For both measurement, they have been calculated and shown in tables below. Table 4.4 and table 4.5 show the mean of time taken every hand measurement for manual measurement and vision measurement respectively.

Table 4.4: Mean of time taken every hand measurement for manual measurement

Total Subject	Finger Length		Palm Length		Palm Breadth		Finger vs Palm (Shape) Index	
	Sum of time (s)	Mean Time (s)	Sum of time (s)	Mean Time (s)	Sum of time (s)	Mean Time (s)	Sum of time (s)	Mean Time (s)
30	343.22	11.4407	334.29	11.1430	266.28	8.876	297.62	9.9207

Table 4.5: Mean of time taken every hand measurement for vision measurement

Total Subject	Finger Length		Palm Length		Palm Breadth		Finger vs Palm (Shape) Index	
	Sum of time (s)	Mean Time (s)	Sum of time (s)	Mean Time (s)	Sum of time (s)	Mean Time (s)	Sum of time (s)	Mean Time (s)
30	143.22	4.7740	52.50	1.7500	135.42	4.5140	22.36	0.7453

4.4.1.1 Analysis on average mean of time taken on both measuring method

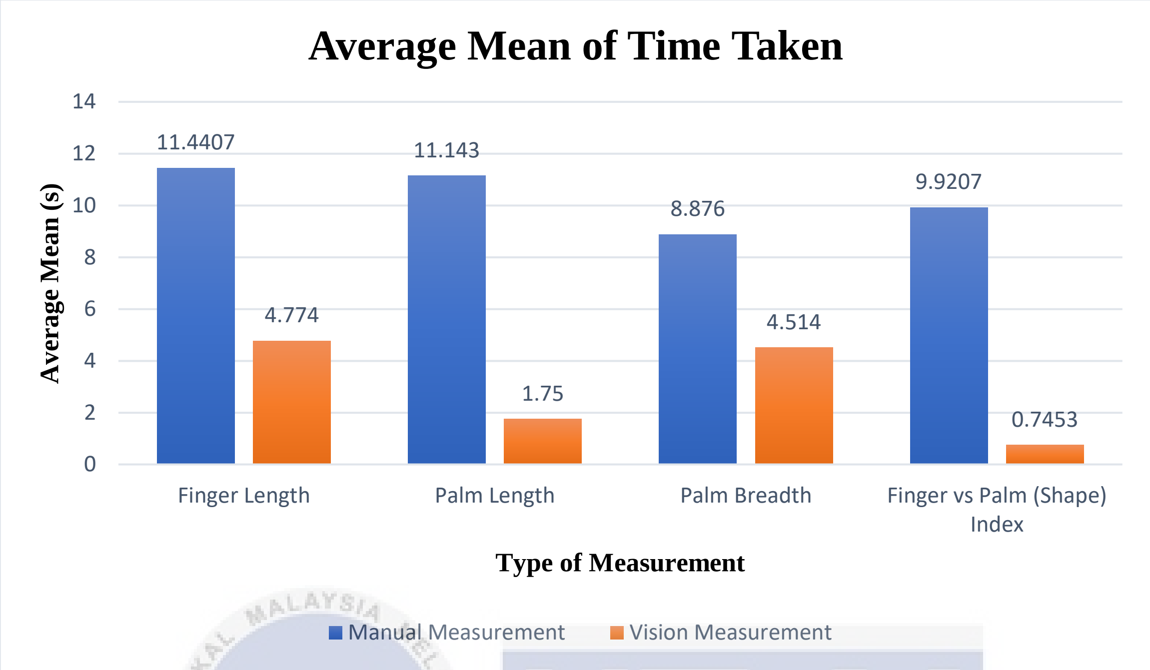


Figure 4.15: Histogram of average mean time taken on both measuring method

The data tabulated in table 4.4 and 4.5 are being transferred and compared using a histogram. Figure 4.15 shows the histogram of average mean time taken on both measuring method. Based on the histogram, vision measurement shows smaller average mean value for every type of measurement. It is obvious that manual measurement needs an average of longer duration to complete a measurement if compare to vision measurement.

Palm length depicts the greatest difference in both manual and vision reading. The time needed can hugely reduce from 11.1430 seconds for manual reading to only 1.750seconds for vision reading. It saves a total of 9.3930 seconds on taking palm length. Finger vs palm (shape) index becomes the second highest difference in time taken because it has a big difference of 9.1754seconds on measuring. Manual measurement and vision measurement for an average of single measuring take 9.9207seconds and 0.7453seconds respectively. For finger length, manual measurement requires 11.4407 seconds but vision measurement needs less than half of it to complete the data, which is 4.7740seconds. Lastly, manual reading takes 8.8760seconds and vision reading takes 4.514seconds to measure palm breadth.

According to this histogram, it shows an inconsistency of reduction in time taken for type of measurement. Assumption made to suggest this scenario is due to the steps in getting the measurement in the source code. If longer step requires in getting the specific type of measurement, the longer time it needs to generate or vice visa. This is also show that the step for finger vs palm(shape) index is the shortest and fastest to complete while palm breadth has the longest step and it requires the longest time to complete the it.

To conclude, the result of mean average in histogram shows that vision measurement takes shorter time in hand measuring compare to manual measurement.

4.4.2 Analyzed and compared two type of hand measuring methods

Bland-Altman plot, also known as Difference plot is a very powerful analytical data plotting method specially designed to analyze the agreement between two different instrument/method provided the methods are applied to the same property. In this project, manual measuring method and vision measuring are two different methods used to measure the same group 30 subjects. Bland-Altman plot focus on the differences exclusively instead of using summarize data to involve correlation and regression.

In hand measuring methods, 4 measurements were recorded and every single type of measurement requires an analysis on agreement to evaluate its value with vision measurement. Therefore, four types of measurement will be compared and Bland-Altman approach is used to analyze every type of measurement. Bland-Altman contains 3 sub parts. The first part is to construct a table and calculate the difference between two paired measurements and average mean of these measurements for every subject. This table should include bias, standard deviation(STD), lower limit of agreement (lower LOA) and upper limit of agreement (upper LOA). Bias is the average of the total difference. Standard deviation(STD) is calculated based on the difference value. The formula for lower LOA = Bias – (1.96*STD). The formula for upper LOA = Bias + (1.96*STD). Compute a regression line graph. This regression line graph helps to determine the relationship between measurements. Lastly, construct Bland-Altman plot. This plot shows the data whether there is any dissimilarity between two set of data and judge if the measurements are comparable.

4.4.2.1 Finger length

Data of finger length will be extracted out from their respective tables and formed a new table. The failure result of vision measurement is eliminated because it will disturb the overall Bland-Altman approach and displayed a very wrong output. Hence, the total data use to carry out Bland-Altman analysis is twenty-four instead of thirty. Table 4.6 shows the data of an agreement between two methods for finger length.

Table 4.6: Data of an agreement between two methods for finger length

Subject	Manual Measurement (mm)	Vision Measurement (mm)	Mean (mm)	Difference (mm)
1	82.00	81.68	81.840	0.32
2	67.50	67.00	67.250	0.50
3	79.00	79.03	79.015	-0.03
5	80.00	81.03	80.515	-1.03
6	69.00	68.88	68.940	0.12
7	71.50	71.00	71.250	0.50
8	80.00	81.09	80.545	-1.09
9	79.50	79.56	79.530	-0.06
12	71.60	70.91	71.255	0.69
13	67.00	67.06	67.030	-0.06
14	71.30	71.33	71.315	-0.03
16	75.10	75.17	75.135	-0.07
17	66.50	66.37	66.435	0.13
18	74.50	74.25	74.375	0.25
19	69.50	69.50	69.500	0.00
20	70.90	70.90	70.900	0.00
21	72.00	72.09	72.045	-0.09
22	73.00	72.10	72.550	0.90
23	70.30	70.00	70.150	0.30
24	76.50	75.50	76.000	1.00
26	82.00	82.06	82.030	-0.06
27	75.00	75.07	75.035	-0.07
28	87.60	87.60	87.600	0.00
29	82.00	82.00	82.000	0.00
		Bias		0.09
		Standard Deviation (STD)		0.4739
		Lower Limit of Agreement (Lower LOA)		-0.8404
		Upper Limit of Agreement (Upper LOA)		1.0171

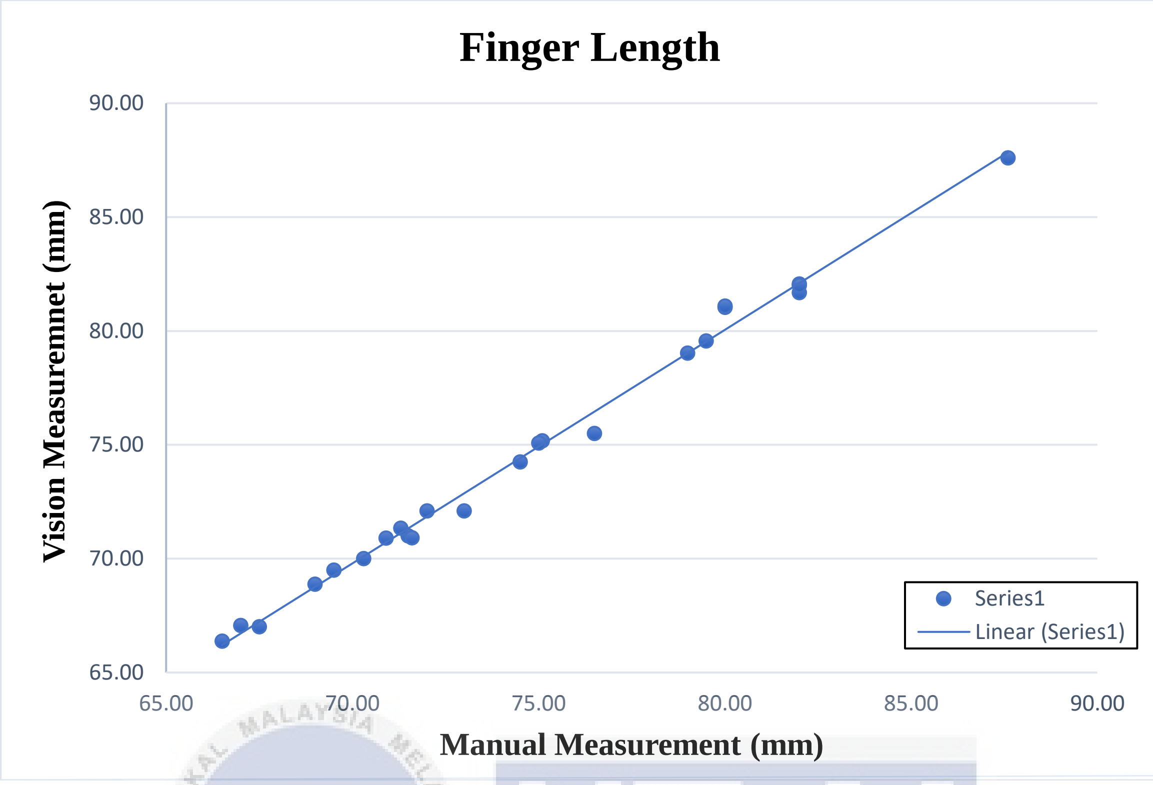


Figure 4.16: Regression line graph of finger length

Figure 4.16 shows regression line graph of finger length. Generally, this linear regression graph attempts to model the significant association between finger length data obtained from both measurements. Refer to the graph, regression line is "Linear (Series 1)" and all finger length data for both measurements are fit to the graph using scatter plot, as plotted in the graph as "Series 1".

This graph shows all dots are located very close to the regression line. Besides, the scatterplot indicates increasing trend along with the regression line. This appears to have a strong strength of relationship between these two measurements. The conclusion can be drawn from this regression graph is the measurements are comparable because they are tightly scattered about the regression line.

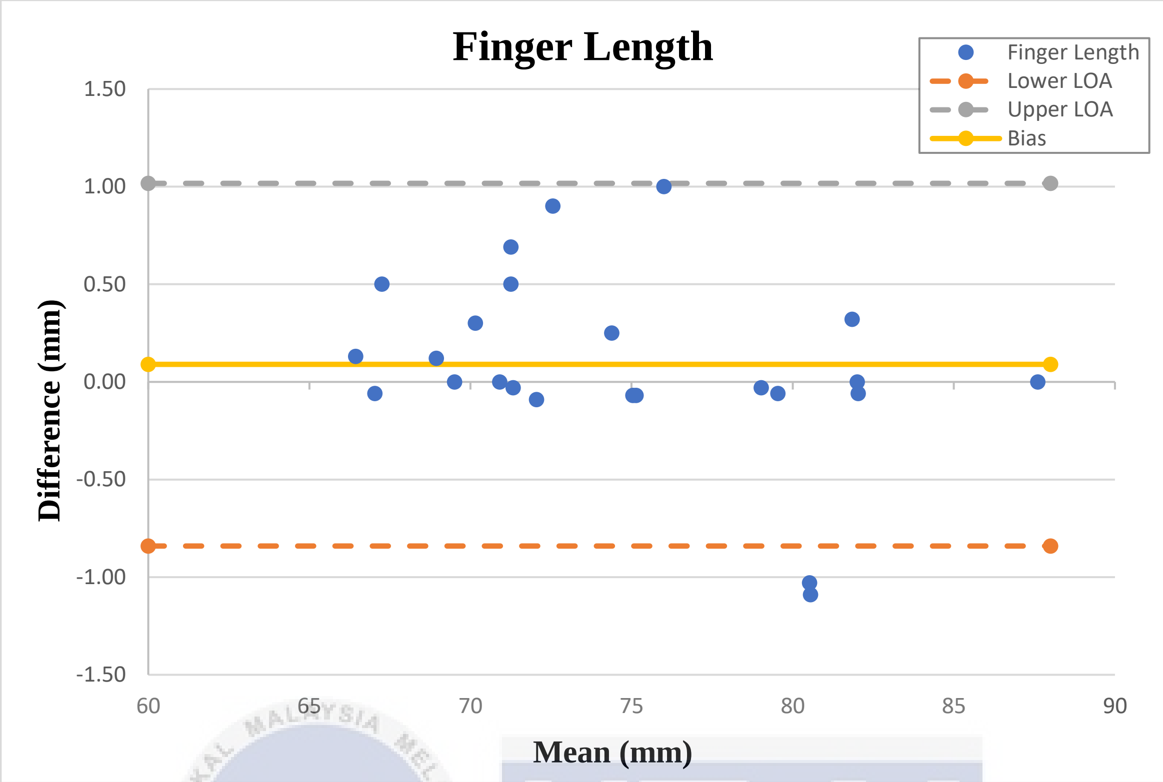


Figure 4.17: Bland Altman plot for finger length

Figure 4.17 shows Bland Altman plot for finger length. This graph is used to quantify the bias and a range of agreement to depict any possible error found. The range of agreement represents the maximum acceptable differences to test the data for any exceeded limits dots.

Based on figure 4.17, the plots have no patterns. However, there is a high concentration within the range of mean from 70mm to 75mm. There are 2 dots out of the lower limit of agreement and one dot is on the line of upper limit of agreement.

To conclude, these two measurements for finger length are comparable. Majority differences of scatter points are within the limit of agreement and gather near to the centered line, which is bias line. This shows a low level of systematic variation with mean of the measurement pairs. Since vision measurement is still under developing progress, perhaps improve it will be the best idea. This is because there are two finger length dots locate outside the area of limit of agreement. These two dots should have located within the limit of agreement.

4.4.2.2 Palm length

Data of palm length will be extracted out from their respective tables and formed a new table. The failure result of vision measurement is eliminated because it will disturb the overall Bland-Altman approach and displayed a very wrong output. Hence, the total data use to carry out Bland-Altman analysis is twenty-four instead of thirty. Table 4.7 shows the data of an agreement between two methods for palm length.

Table 4.7: Data of an agreement between two methods for palm length

Subject	Manual Measurement (mm)	Vision Measurement (mm)	Mean (mm)	Difference (mm)
1	112.00	111.08	111.540	0.92
2	88.50	88.50	88.500	0.00
3	99.00	99.03	99.015	-0.03
5	99.00	98.86	98.930	0.14
6	89.00	89.00	89.000	0.00
7	91.00	90.79	90.895	0.21
8	97.00	105.66	101.330	-8.66
9	99.50	99.50	99.500	0.00
12	95.80	95.60	95.700	0.20
13	97.30	97.00	97.150	0.30
14	105.60	105.60	105.600	0.00
16	91.40	91.00	91.200	0.40
17	93.50	93.30	93.400	0.20
18	99.70	99.59	99.645	0.11
19	94.90	94.90	94.900	0.00
20	91.60	91.66	91.630	-0.06
21	102.00	102.05	102.025	-0.05
22	91.50	91.42	91.460	0.08
23	94.20	94.00	94.100	0.20
24	99.80	99.67	99.735	0.13
26	112.40	112.35	112.375	0.05
27	93.00	93.00	93.000	0.00
28	109.30	109.25	109.275	0.05
29	118.00	118.03	118.015	-0.03
		Bias		-0.24
		Standard Deviation (STD)		1.8046
		Lower Limit of Agreement (Lower LOA)		-3.7804
		Upper Limit of Agreement (Upper LOA)		3.2937

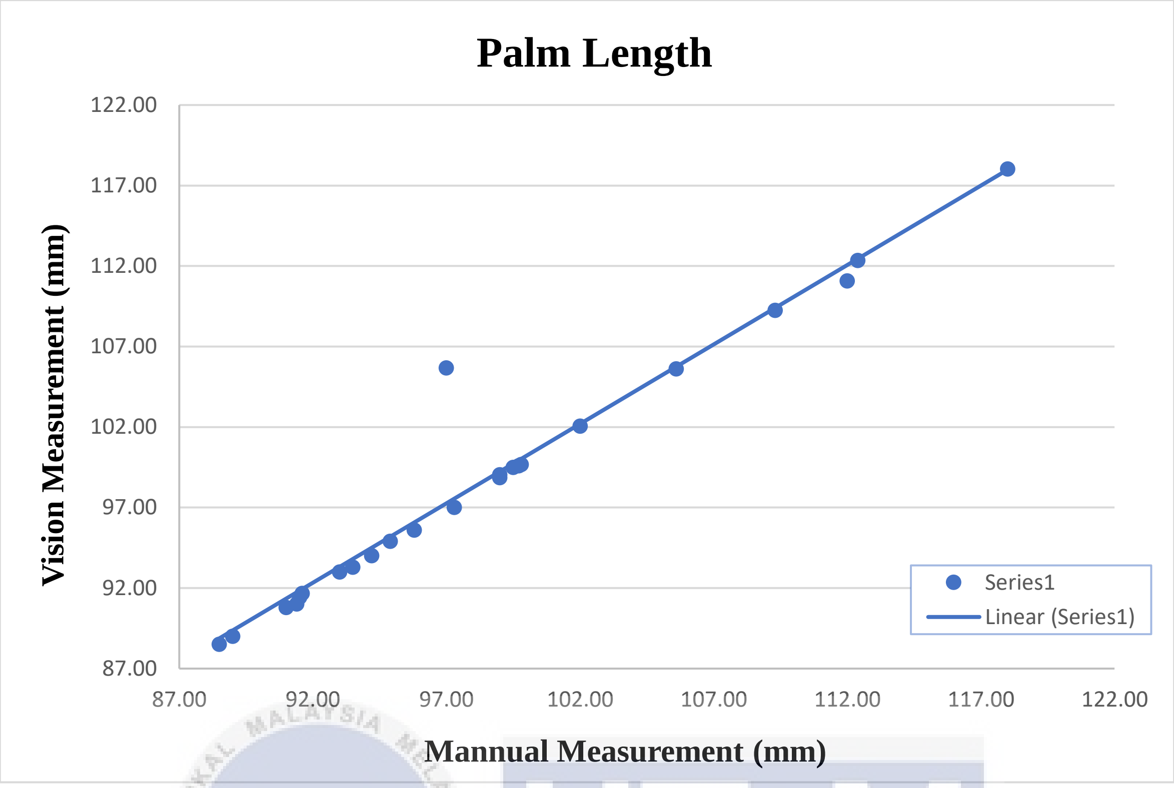


Figure 4.18: Regression line graph of palm length

Figure 4.18 shows regression line graph of palm length. In short, this linear regression graph functions to model the significant association between palm length data obtained from both measurements. Refer to the graph, regression line is "Linear (Series 1)" and all palm length data for both measurements are fit to the graph using scatter plot, as plotted in the graph as "Series 1".

This graph shows all dots are located very close to the regression line and a disproportionate influential point that diverge a huge way from the overall scatter plot pattern. This influential point is an outliner distant from the rest data without extreme x and y. If ignore the outliner, the rest scatter points look satisfactory and increase along the regression line trend. This represents the data have a strong strength of relationship between these two measurements. The conclusion can be drawn from this regression graph is the measurements are comparable because they are tightly scattered about the regression line. However, the appearance of an outliner represents bad data had collected, possibly the result of measurement error. A brief checking data on difference column using table 4.7 found out that the main error was occurred on data subject 8. The measurement for both methods show a big difference of 8.66 but it cannot conclude which method caused this error.

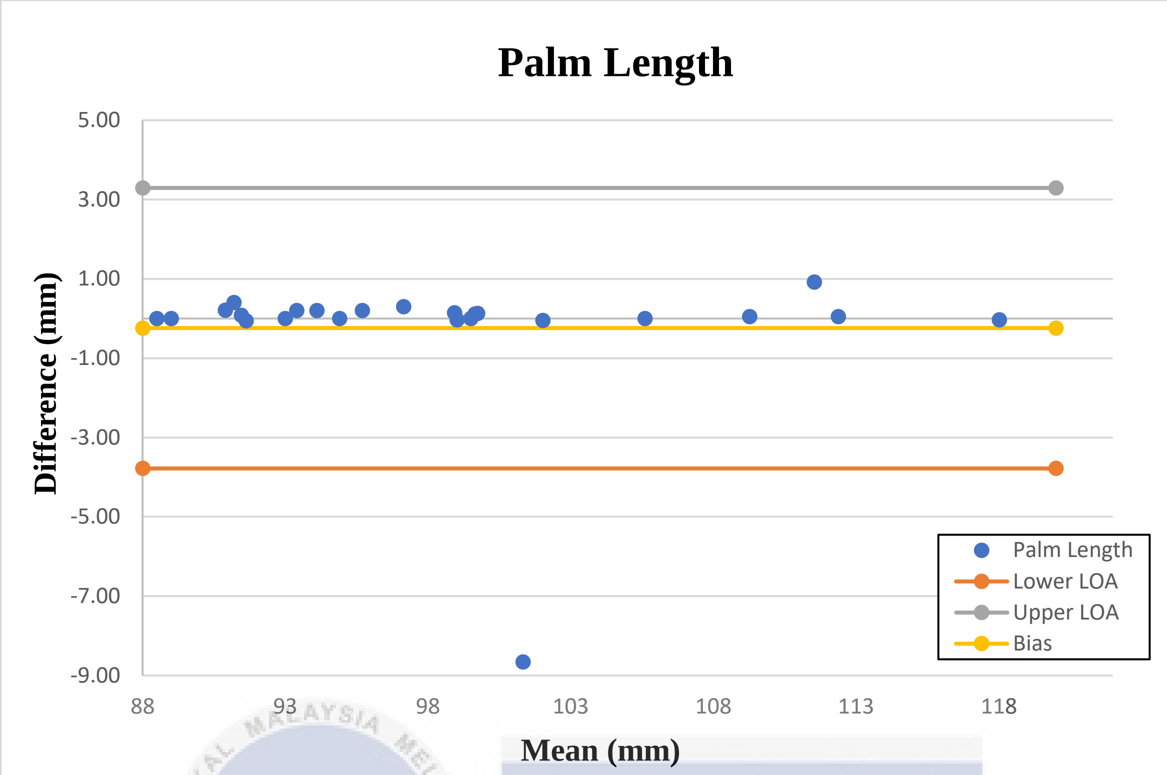


Figure 4.19: Bland Altman plot for palm length

Figure 4.19 shows Bland Altman plot for palm length. This graph is used to quantify the bias and a range of agreement to depict any possible error found. The range of agreement represents the maximum acceptable differences to test the data for any exceeded limits dots.

Based on figure 4.19, the plots have linear pattern along the bias line. There is a high concentration within the range of mean from 90mm to 100mm. There is a dot located out of the lower limit of agreement.

To conclude, these two measurements for palm length are comparable. If the outliner is not counted, the rest of the dots are within the limit of agreement and have a small distance from the bias centered line. If without the outliner, it shows no systematic variation with mean of the measurement pairs. Since vision measurement is still under developing progress, perhaps improve it will be the best idea. This is because one dot is found outside the area of limit of agreement. It supposes to locate within the limit of agreement.

4.4.2.3 Palm breadth

Data of palm breadth will be extracted out from their respective tables and formed a new table. The failure result of vision measurement is eliminated because it will disturb the overall Bland-Altman approach and displayed a very wrong output. Hence, the total data use to carry out Bland-Altman analysis is twenty-four instead of thirty. Table 4.8 shows the data of an agreement between two methods for palm breadth.

Table 4.8: Data of an agreement between two methods for palm breadth

Subject	Manual Measurement (mm)	Vision Measurement (mm)	Mean (mm)	Difference (mm)
1	95.00	93.01	94.005	1.99
2	75.00	73.50	74.250	1.50
3	71.50	71.00	71.250	0.50
5	79.00	77.30	78.150	1.70
6	67.50	66.20	66.850	1.30
7	69.50	70.00	69.750	-0.50
8	72.00	91.23	81.615	-19.23
9	78.00	77.66	77.830	0.34
12	70.60	70.00	70.300	0.60
13	69.00	68.31	68.655	0.69
14	71.20	71.11	71.155	0.09
16	74.50	73.50	74.000	1.00
17	74.90	72.90	73.900	2.00
18	76.60	76.00	76.300	0.60
19	74.30	74.38	74.340	-0.08
20	70.40	70.40	70.400	0.00
21	68.00	68.00	68.000	0.00
22	74.00	73.55	73.775	0.45
23	75.15	74.99	75.070	0.16
24	74.20	73.01	73.605	1.19
26	88.30	87.88	88.090	0.42
27	78.40	78.10	78.250	0.30
28	85.00	85.00	85.000	0.00
29	91.00	90.36	90.680	0.64
		Bias		-0.18
		Standard Deviation (STD)		4.1123
		Lower Limit of Agreement (Lower LOA)		-8.2409
		Upper Limit of Agreement (Upper LOA)		7.8792

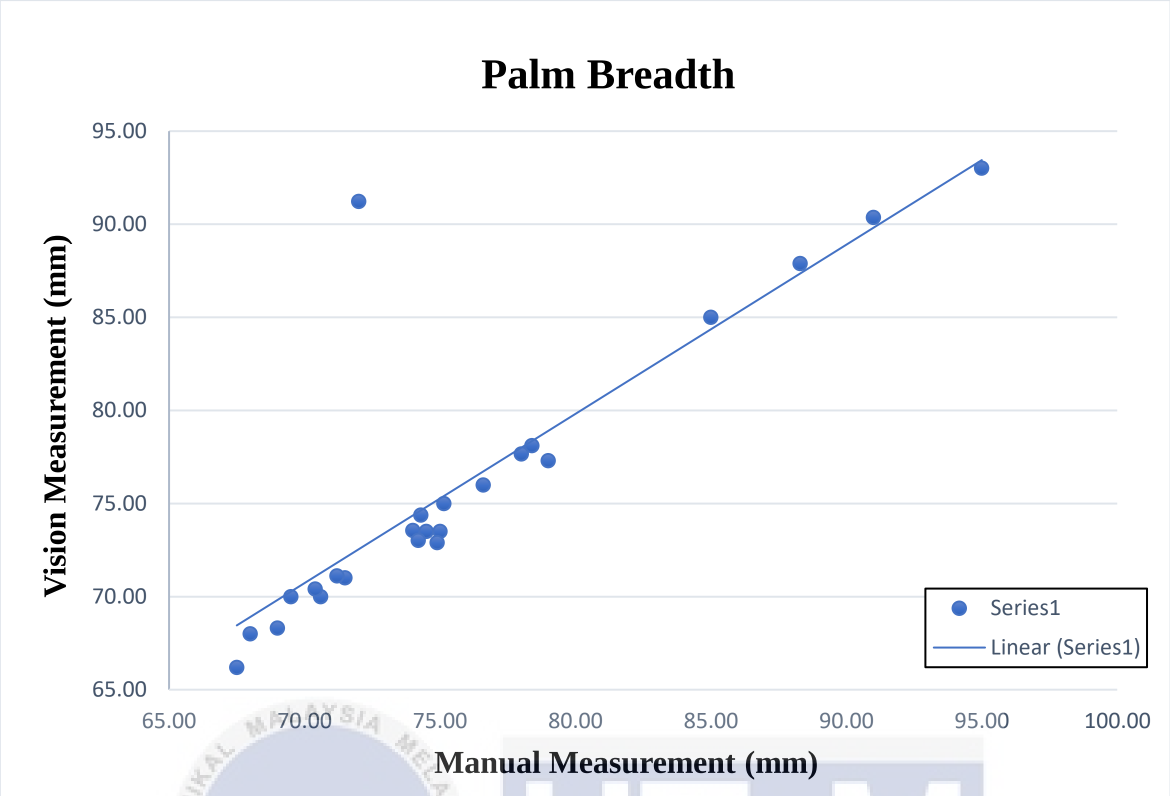


Figure 4.20: Regression line graph of palm breadth

Figure 4.20 shows regression line graph of palm breadth. Basically, this linear regression graph is used to model the significant association between palm breadth data obtained from both measurements. Refer to the graph, regression line is "Linear (Series 1)" and all palm breadth data for both measurements are fit to the graph using scatter plot, as plotted in the graph as "Series 1".

This graph shows all dots tightly scattered about the regression line and a left out influential point that diverge a huge way from the overall scatter plot pattern. This influential point is an outliner. If ignore the outliner, the rest scatter points look satisfactory and increase along the regression line trend. This shows the data have a strong relationship between these two measurements. The conclusion can be drawn from this regression graph is the measurements are comparable because they are tightly scattered about the regression line. However, the appearance of an outliner represents bad data had collected, possibly the result of measurement error. A brief checking data on difference column using table 4.8 found out that the main error was occurred on data subject 8. This is the second-time subject 8 data has error. The measurement for both methods show a big difference of 19.23 but it cannot determine which method caused this error.

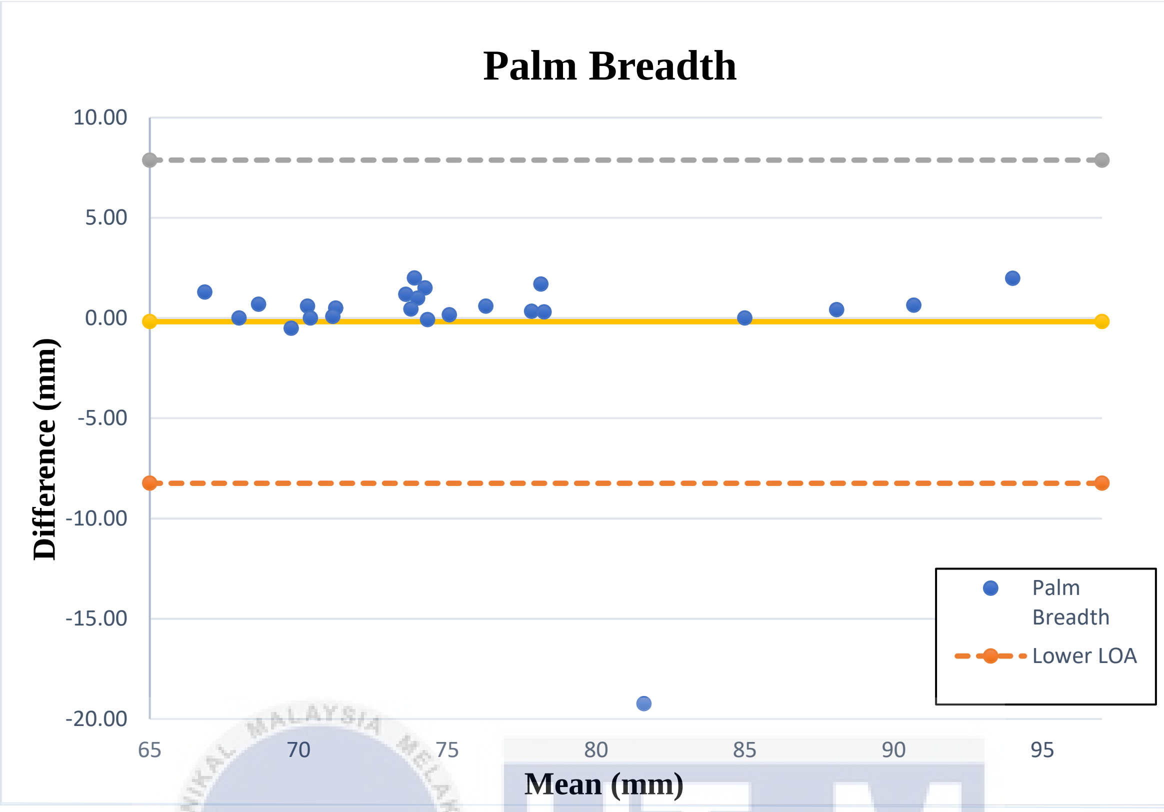


Figure 4.21: Bland Altman plot for palm breadth

Figure 4.21 shows Bland Altman plot for palm breadth. This graph is used to quantify the bias and a range of agreement to depict any possible error found. The range of agreement represents the maximum acceptable differences to test the data for any exceeded limits dots.

Based on figure 4.21, the plots have linear pattern all position above the bias line. There is a high concentration within the range of mean from 73mm to 79mm. There is a dot located out of the lower limit of agreement.

To conclude, these two measurements for palm breadth are comparable. If the outliner is not counted, the rest of the dots are within the limit of agreement and have a small distance from the bias centered line. If without the outliner, it shows no systematic variation with mean of the measurement pairs. Since vision measurement is still under developing progress, perhaps improve it will be the best idea. This is because one dot is found outside the area of limit of agreement. It supposes to locate within the limit of agreement.

4.4.2.4 Finger vs palm (shape) index

Data of finger vs palm (shape) index will be extracted out from their respective tables and formed a new table. The failure result of vision measurement is eliminated because it will disturb the overall Bland-Altman approach and displayed a very wrong output. Hence, the total data use to carry out Bland-Altman analysis is twenty-four instead of thirty. Table 4.9 shows the data of an agreement between two methods for finger vs palm (shape) index.

Table 4.9: Data of an agreement between two methods for finger vs palm (shape) index

Subject	Manual Measurement (mm)	Vision Measurement (mm)	Mean (mm)	Difference (mm)
1	39.61	40.02	39.818	-0.41
2	41.28	41.36	41.321	-0.07
3	46.33	46.48	46.407	-0.15
5	44.94	46.00	45.471	-1.05
6	44.09	44.38	44.235	-0.29
7	44.55	44.16	44.353	0.39
8	47.34	41.19	44.261	6.15
9	44.79	44.91	44.849	-0.12
12	43.03	42.82	42.924	0.21
13	40.29	40.57	40.427	-0.28
14	40.33	40.37	40.347	-0.04
16	45.27	45.70	45.482	-0.43
17	39.49	39.93	39.712	-0.44
18	42.26	42.29	42.272	-0.03
19	41.08	41.06	41.066	0.02
20	43.77	43.75	43.757	0.02
21	42.35	42.39	42.373	-0.04
22	44.11	43.70	43.907	0.40
23	41.51	41.42	41.467	0.09
24	43.97	43.72	43.844	0.24
26	40.86	40.98	40.920	-0.13
27	43.76	43.87	43.816	-0.12
28	45.08	45.10	45.091	-0.01
29	39.23	39.35	39.292	-0.11
		Bias		0.16
		Standard Deviation (STD)		1.3118
		Lower Limit of Agreement (Lower LOA)		-2.4126
		Upper Limit of Agreement (Upper LOA)		2.7296

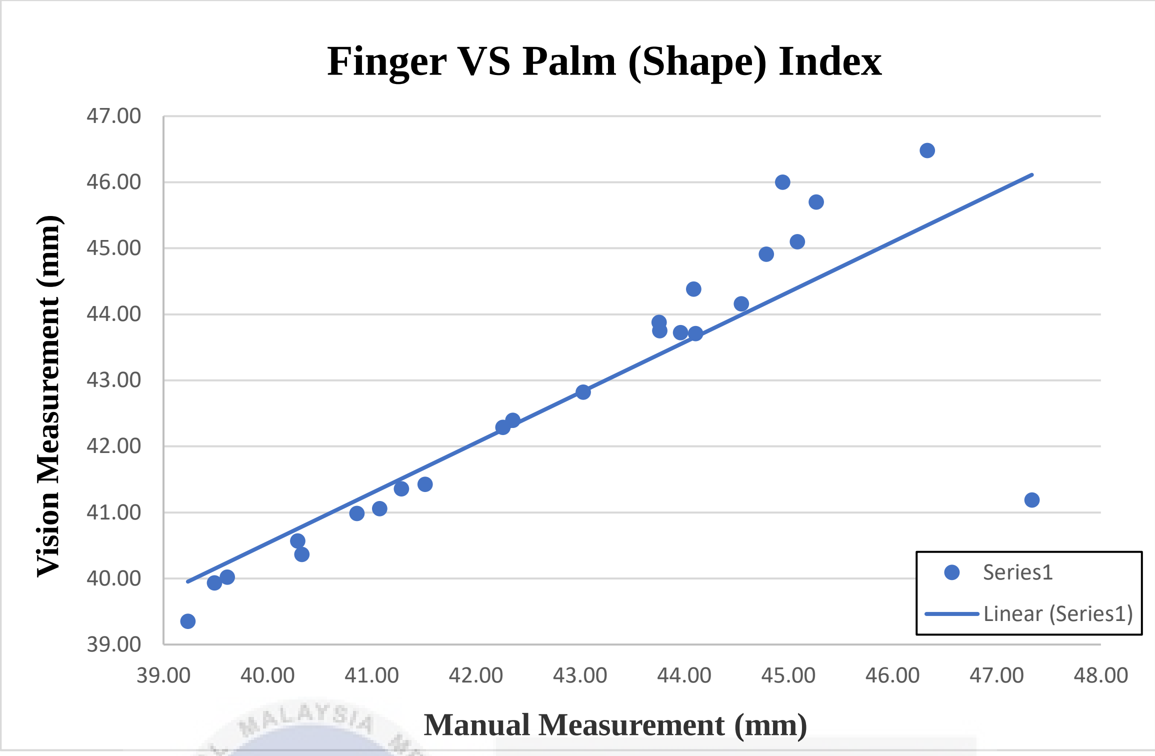


Figure 4.22: Regression line graph of finger vs palm (shape) index

Figure 4.22 shows regression line graph of finger vs palm (shape) index. Linear regression graph models the significant association between finger vs palm (shape) index data obtained from both measurements. Refer to the graph, regression line is "Linear (Series 1)" and all finger vs palm (shape) index data for both measurements are fit to the graph using scatter plot, as plotted in the graph as "Series 1".

This graph shows all dots are loosely scattered about the regression line but still within the acceptable line range. Only one influential point is out of range because it diverges very far away from the scatter plot pattern. Influential point also known as outliner. This outliner could have an extreme x value. If ignore this outliner, the rest scatter points look satisfactory and increase along the regression line trend. This data shows connection between these two measurements. Although an outliner is found, this regression graph is still achieving agreement determination because the scattered points are scattered about the regression line. Having bad data collected is mostly likely the result of measurement error. A brief checking data on difference column using table 4.9 found out that the main error was occurred on data subject 8. Subject 8 has caused error for every measurement except finger length. The measurement for both methods show a big difference of 6.15 but it cannot determine which method caused this error.

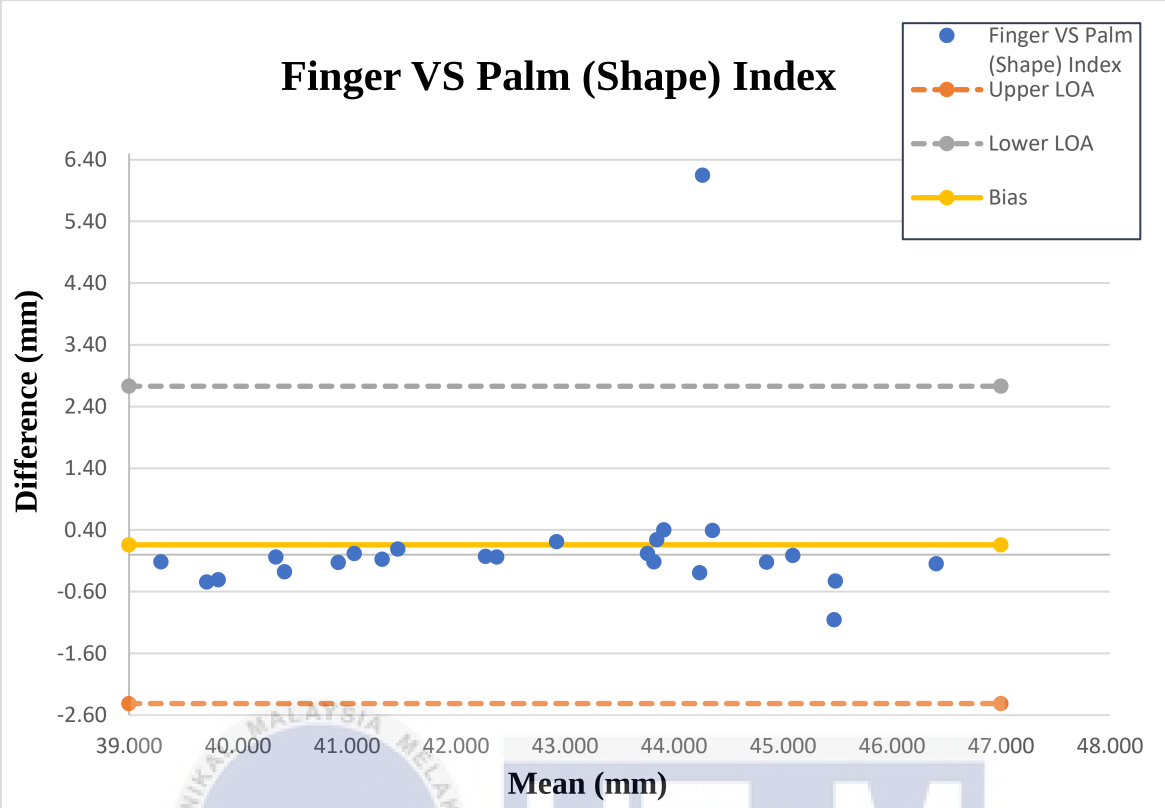


Figure 4.23: Bland Altman plot for finger vs palm (shape) index

Figure 4.23 shows Bland Altman plot for finger vs palm (shape) index. The graph has plots with linear pattern along the bias line. A dot is located out of the upper limit of agreement. To conclude, these two measurements for finger vs palm (shape) index are comparable. If the outliner is not counted, the rest of the dots are within the limit of agreement it shows no systematic variation with mean of the measurement pairs. Since vision measurement is still under developing progress, perhaps improve it will be the best idea. This is because one dot is found outside the area of limit of agreement. It supposes to locate within the limit of agreement.

4.5 SUMMARY

To sum up, programming is developed and attained completely the first objective. Data collection for thirty subjects shows there are six failures caused by drawing contour function in vision measurement. Vision measurement requires shorter time taken for measurement compare to manual measurement. Both data collection using manual and vision measurement are verified to have strong correlation and agreement between them.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Measurement system plays an important role to provide unit to outline the identity and characteristic of a product. The unit in SI unit standard can be either inch, feet, meter, millimeter, centimeter, kilometer or other related measurement unit depends on the size and purpose of the product. Anthropometry data is related to almost every industry but this project only focus on human hand palm measurement. In conjunction with measuring system, it is essential to upgrade it to vision measuring system to replace manual measuring system which has less efficiency and disadvantage which had been outline in chapter 2.

The 1st objective is to develop a measuring system for hand palm measurement. Python with library OpenCV is used as main program to develop the whole script and run as a platform to measure hand palm instead of using a ready-made software such as Matlab. This image processing coding only focus on hand breadth, hand palm and finger length measurement. Finger vs palm (shape) index is generated using the collected measurement based on the formula given. The user's image is captured using Picamera and data transmitted to Raspberry Pi controller for image processing development and displayed on laptop screen. This objective is successfully achieved.

2nd objective is to validate vision sensor measurement method with manual measurement method in terms of time consumption. An exactly thirty subjects are collected manually and using vision method. However, offline method was applied for vision method due to time constraint. Both data are tabulated for comparison and analysing. For manual measurement, the average mean of time taken for finger length, palm length, palm breadth and finger vs palm (shape) index are 11.4407 seconds, 11.143 seconds, 8,876 seconds and

9.9207 seconds respectively. For vision measurement, the average mean of time taken for finger length, palm length, palm breadth and finger vs palm (shape) index are 4.774 seconds, 1.75 seconds, 4.514 seconds and 0.7453 seconds respectively. Hence, vision method is proved to faster in measuring. This objective is achieved.

3rd objective is to compare vision sensor measurement method with manual measurement method. Same thirty collected data used for objective 2 is applied for this objective. The result shows an error for subject number 8 but the rest result comparison using linear regression graph shows the both measurements are comparable because they are tightly scattered about the regression line. As a result, this objective is achieved.

In conclusion, vision measurement method can replace manual measurement method to collect the anthropometry data. The objectives and scopes are fully accomplished. The following part in this chapter will briefly outline the bill of materials (BOM), sustainable design and development, complexity, long life learning (LLL) and basic entrepreneurship (BE) as well as limitations for this vision measuring system and recommendations and suggestions to minimize the limitation to continuing enhance this system.

5.2 BILL OF MATERIAL (BOM)

A summary of bill of material on table 5.1 shows the overall price to carry out this research. This bill of material is only for reference because the price of the part is varied depends on the method of buying, such as online purchasing or on shop buying.

Table 5.1: Bill of material (BOM)

Number	Part	Quantity	Price (RM)
1	Laptop	1	2500.00
2	Controller (Raspberry Pi)	1	164.99
3	Camera (Raspberry Pi camera)	1	32.00
4	Micro USB cable	1	5.00
5	Ethernet cable	1	8.00
6	Micro SD card	1	35.00
7	Universal memory card reader	1	16.00
Total			2760.99

5.3 SUSTAINABLE DESIGN AND DEVELOPMENT

A sustainable design product has five basic principles which are cyclic, solar, efficient, safe and social. For this project, these five basic principles have achieved. Figure 5.1 shows an example of Raspberry Pi life cycle.

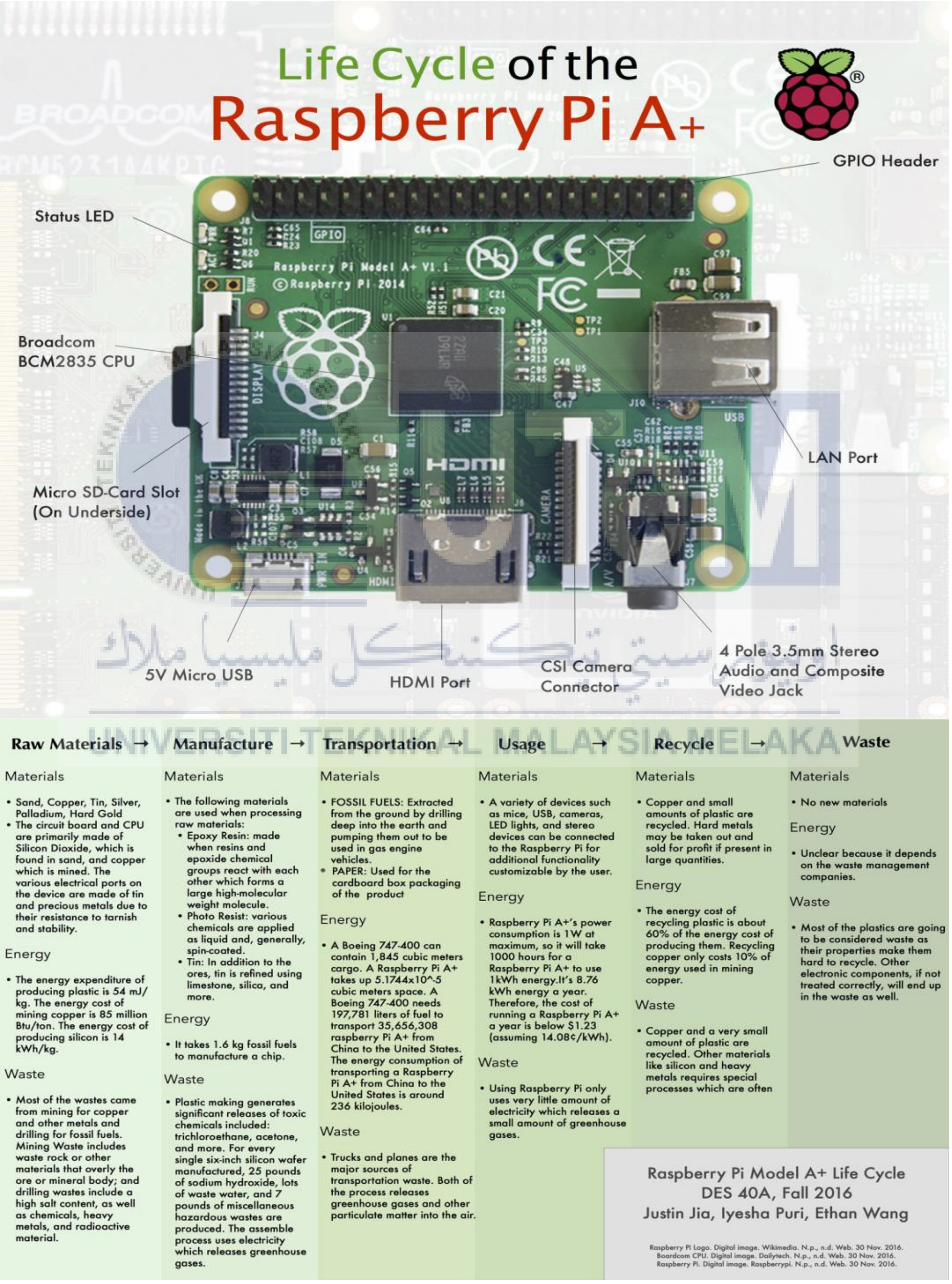


Figure 5.1: Life cycle of Raspberry Pi
(http://www.designlife-cycle.com/raspberry-pi/)

The life cycle of Raspberry Pi shows that the primary stage is from raw material acquisition and manufacturing but the subsequent processes of distribution, use, recycling, and waste management introduce the importance of environmentally friendly and new products. For the use of laptop in this project, after the laptop is stripped down with the components removed, such as all the batteries removed and metal frame dismantled, the rest of the parts can be recycled. This shows that a cyclic manufacturing of these product due to recycle and then reuse and remanufacturing again. Since the parts can be remanufactured and cyclic within a closed loop, it is considered as solar. Basically, this vision measurement system is efficient because Raspberry Pi is eco-friendly and energy consumption lesser than other computer devices. Raspberry Pi predecessor delivers stronger performances yet consumes less power. It is safe to used and a proper disposal of computer devices can avoid the pollution. Lastly, the components used in this project accomplish social because it supports basic natural justice by introducing recycling to protect the environment.

To conclude, environmental conscious manufacturing (ECM) which emphasized on application of cleaner and more efficient manufacturing methods is accomplish with the used of Raspberry Pi components. The main reason is due to its relatively low energy consumption and use more environmentally preferable material to protect environment as well as recycling and reusing the parts. The overall cost is affordable for system built up and hence it is economic friendly.

5.4 COMPLEXITY

This project attains complex engineering problems because the Python programming development required critical thinking and patience to justify which part of hand palm to measure and how to make indication using image processing. A better mathematical theory and methods applied. All the stages involved problem solving and techniques have been chosen wisely to increase the probability of getting a comparable result with manual measuring method. These significant findings for the result encompass and accomplish the complex problem solving, complex engineering problems and knowledge profile on the background of techniques use.

5.5 LONG LIFE LEARNING (LLL) AND BASIC ENTREPRENEURSHIP (BE)

This project promotes long life learning (LLL) because it requires self-study to develop the complete programming measuring system. Long life learning takes time for personal development because Python programming knowledge is very wide without a border and it can be very complex as well as competitiveness. Throughout the project, some ideas develop and can be used for further improvement. The overall vision system can be developed without laptop. This is because laptop is function as a screen to display the internal environment of the Raspberry Pi. To be eco-friendly by decreasing the energy consumption, a screen purchasing should be done to replace the laptop. To enhance the system, the idea of improving the programming should also be done to reduce the error.

If all the ideas stated can be achieve, the next step would be make this system more user-friendly, perhaps Graphic-User-Interface (GUI) application and a portable size will make it more potential to be commercialized. A further study to make this system become a phone application might have a higher chance to attract the potential buyer.

5.6 LIMITATION

Nothing in the world is 100% perfect without limitations. In this project, although all the objectives are fully achieved, they still have limitations which cannot be eliminated but only minimize to avoid causing serious defect to the system. The limitations are as follows:

1. Camera calibration

   The camera calibration is difficult to set in the proper position with correct length and angle because a small angle can cause the image captured to be out of expected position. This small angle is difficult to detect with human eye during setup and there is no proper mechanical structure to hold the camera due to money and time constraint. An incorrect spot for capturing image can cause the hand cannot be detected and measurement cannot be taken.

2. Lighting

   This python image processing is first remove all the noise but the shape of the hand detected is affected by the degree of light reflect on the black background. This black background is used to put the user's hand. A strong reflection can direct a wrong contour detection and the hand shape can be wrongly detected. On the other hand, if the light intensity in the environment is low, means it is too dark, the same wrong contour detection can occur. To carry out this measurement in a room with the ceiling lamp can minimize this problem.

3. Image processing

   The critical part when running the image processing script is to detect the shape of hand and locate the correct point on hand for further measurement. On the image, hand is considering as the smallest area detected. Sometimes, the function of contour detection will detect something else such as a light reflection dot on the black board and cause the rest of the program can be run because no hand was detected. This is the main issue the percentage of vision reading cannot achieve 100% but 80% with 20% failure.

4. Offline testing for verification

   Offline testing was used to verify the validity measurement of vision with the reference of manual measurement data. This method involved hand image capturing using the same Picamera but the images were stored until the image processing was fully developed and ready for verification. In short, the verification was using image instead of on the spot image captured and processed to gain the measurement. This is due to the time constraint to complete the whole project and it can cause the bugs for the online image processing to be left out.

5.7 RECOMMENDATIONS AND SUGGESTIONS

This vision measuring method certainly can be improved to higher level to provide end-line service to the users eventually. There are many recommendations and suggestions can be applied to upgrade tis system to become more stable by minimize or even eliminate the discussed limitations.

1. Mechanical structure

   A proper setup mechanical structure is the tool to minimize the camera calibration problem. This mechanical structure should have the ability to adjust its height and housing a small camera and place the controller. Preferably, the material is made from ABS using 3D printing. This can in turns provide a light material setup structure which can hold the camera in the position to capture image.

2. Programming

   Percentage of failure due to the programming is consider high. Further research and analysis should be carry out to improve the written script and implement a more stable and reliable programming.

3. Attached lighting system

   An internal attached lighting system can be done by building a LED attached to a breadboard and using the Raspberry Pi controller to control it. This all can be run by adding the LED part to the Python script and control the lighting duration. It will spark a flash when taking the image of hand but its degree of lighting need to be controlled to avoid strong reflection on the black background. This can resolve the problem of only carry out in a room with ceiling lamp. This also means that hand measurement system can carry out in any location regardless of the environment if lighting system develops.

4. User-friendly graphical-user-interface (GUI)

   The whole system was written in Python language and run using Raspberry Pi terminal to output the result of the measurement. It is not very practical because the public who cannot read computer language cannot use it. Hence, it is highly recommended to build out a user-friendly screen which provide graphical-user-interface where the language used is human-understandable-common language.

5. Internet connection

   This system can only function when there is internet connection. This is due to the reason Raspberry Pi controller needs internet to function. It can become a serious problem when the local internet connection is weak. Perhaps changing its mother controller to Arduino board will be more practical.

REFERENCES

Agarwal, P., & Sahu, S. (2010). Determination of hand and palm area as a ratio of body surface area in Indian population. *Indian Journal of Plastic Surgery*, *43*(1), 49. https://doi.org/10.4103/0970-0358.63962

Al-amri, S. S. (2010). Image segmentation by using edge detection, *2*(3), 804–807.

Berglund, J., Lindskog, E., Vallhagen, J., Wang, Z., Berlin, C., & Johansson, B. (2016). Production System Geometry Assurance Using 3D Imaging. *Procedia CIRP*, *44*, 132–137. https://doi.org/10.1016/j.procir.2016.02.138

Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: Advances and prospects. *Pattern Recognition*, *34*(12), 2259–2281. https://doi.org/10.1016/S0031-3203(00)00149-7

Chitradevi, B., & Srimanthi, P. (2014). An Overview on Image Processing Techniques. *ISRN Signal Processing*, *2*(11), 6466–6472. https://doi.org/10.1155/2013/496701

Durand-Petiteville, A., Vougioukas, S., & Slaughter, D. C. (2017). Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing. *Computers and Electronics in Agriculture*, *142*, 298–313. https://doi.org/10.1016/j.compag.2017.09.011

Engineering, I. (1990). ON THE COLOR IMAGE SEGMENTATION ALGORITHM BASED ON THE THRESHOLDING AND THE FUZZY c-MEANS TECHNIQUES, *23*(9), 935–952.

Fan, J., Yau, D. K. Y., Elmagarmid, a K., & Aref, W. G. (2001). Automatic image segmentation by integrating color-edge extractionand seeded region growing. *IEEE Transactions on Image Processing*, *10*(10), 1454–1466.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation, *59*(2), 167–181.

Flack, D. (2009). Good Practice Guide No. 80, (80).

Gadelmawla, E. S. (2017). Computer vision algorithms for measurement and inspection of external screw threads. *Measurement: Journal of the International Measurement Confederation*, *100*, 36–49. https://doi.org/10.1016/j.measurement.2016.12.034

Greiner, T. M. (1991). HAND ANTHROPOMETRY OF U . S . ARMY PERSONNEL By, (December 1990).

Haralick, R. M. (1985). Image Segmentation Techniques, 100–132.

Hossein Mirzabe, A., Barati kakolaki, M., Abouali, B., & Sadin, R. (2017). Evaluation of some engineering properties of cucumber (Cucumis sativus L.) seeds and kernels based on image processing. *Information Processing in Agriculture*, *4*(4), 300–315. https://doi.org/10.1016/j.inpa.2017.07.001

Ibrahim, M. A. B., Khalifa, A. M., Hagras, A. M., & Alwakid, N. I. (2016). Sex determination from hand dimensions and index/ring finger length ratio in North Saudi population: Medico-legal view. *Egyptian Journal of Forensic Sciences*, *6*(4), 435–444. https://doi.org/10.1016/j.ejfs.2016.11.002

Jee, S. C., & Yun, M. H. (2016). An anthropometric survey of Korean hand and hand shape types. *International Journal of Industrial Ergonomics*, *53*, 10–18. https://doi.org/10.1016/j.ergon.2015.10.004

Kırcı, M., Güneş, E. O., & Çakır, Y. (2014). Vegetation Measurement Using Image Processing Methods. *The Third International Conference on Agro-Geoinformatics*.

Klonowski, M., Plata, M., & Syga, P. (2018). User authorization based on hand geometry without special equipment. *Pattern Recognition*, *73*, 189–201. https://doi.org/10.1016/j.patcog.2017.08.017

Krechmer, K. (2018). Relative measurement theory ☆ The uni fi cation of experimental and theoretical measurements. *Measurement*, *116*(November 2017), 77–82. https://doi.org/10.1016/j.measurement.2017.10.053

Li, Z., Jia, W., Mao, Z.-H., Li, J., Chen, H.-C., Zuo, W., … Sun, M. (2013). Anthropometric body measurements based on multi-view stereo image reconstruction. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2013*(61173086), 366–9. https://doi.org/10.1109/EMBC.2013.6609513

Maini, R. (n.d.). Study and Comparison of Various Image Edge Detection Techniques, *147002*(3), 1–12.

Mandahawi, N., Imrhan, S., Al-Shobaki, S., & Sarder, B. (2008). Hand anthropometry survey for the Jordanian population. *International Journal of Industrial Ergonomics*, *38*(11–12), 966–976. https://doi.org/10.1016/j.ergon.2008.01.010

Massiris, M., Peña-Baena, R., Oviedo-Trespalacios, & Maestre-Meyer, M. (2015). Hand Anthropometry of Colombian Caribbean College Students Using Software Based Method. *Procedia Computer Science*, *67*(Dsai), 123–131. https://doi.org/10.1016/j.procs.2015.09.256

Min, J., & Principle, A. M. (2015). Comprehensive Measurement System for Screw Thread Parameter Based on Machine Vision, 266–270.

Momin, M. A., Rahman, M. T., Sultana, M. S., Igathinathane, C., Ziauddin, A. T. M., & Grift, T. E. (2017). Geometry-based mass grading of mango fruits using image processing. *Information Processing in Agriculture*, *4*(2), 150–160. https://doi.org/10.1016/j.inpa.2017.03.003

Mutneja, V. (2015). Electrical & Electronic Systems Methods of Image Edge Detection : A Review, *4*(2). https://doi.org/10.4172/2332-0796.1000150

Ning, J., Zhang, L., Zhang, D., & Wu, C. (2010). Interactive image segmentation by maximal similarity based region merging ☆. *Pattern Recognition*, *43*(2), 445–456. https://doi.org/10.1016/j.patcog.2009.03.004

Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, *26*(9), 1277–1294. https://doi.org/10.1016/0031-3203(93)90135-J

Paolo, L., Fruggiero, F., Fera, M., Lambiase, A., Maresca, P., & Caja, J. (2017). ScienceDirect ScienceDirect ScienceDirect ScienceDirect The role Engineering of human fatigue in the Conference uncertainty of measurement Costing for capacity optimization in Industry Trade-off between used capacity and operational efficiency The role of. *Procedia Manufacturing*, *13*, 1320–1327. https://doi.org/10.1016/j.promfg.2017.09.092

Park, S. J., Min, S. N., & Lee, H. (n.d.). 3D Hand Anthropometry of Korean Teenager ' s and Comparison with Manual Method, 491–492.

Parker, J. A., Kenyon, R. V., & Young, L. R. (1985). Measurement of Torsion from Multitemporal Images of the Eye Using Digital Signal Processing Techniques. *IEEE Transactions on Biomedical Engineering*, *BME-32*(1), 28–36. https://doi.org/10.1109/TBME.1985.325613

Parveen, E. Z. (2017). Assessment of Quality of Rice Grain using Optical and Image Processing Technique, 265–270.

Qi, J., Zhang, X., Ying, B., & Lv, F. (2011). Comparison of Human Body Sizing Measurement Data by Using Manual and 3D Scanning Measuring Techniques. *Journal of Fiber Bioengineering and Informatics*, *4*(1), 83–95. https://doi.org/10.3993/jfbi04201109

Sarikan, S. S., Ozbayoglu, A. M., & Zilci, O. (2017). Automated Vehicle Classification with Image Processing and Computational Intelligence. *Procedia Computer Science*, *114*, 515–522. https://doi.org/10.1016/j.procs.2017.09.022

Sharma, N., Ray, A., Shukla, K., Sharma, S., Pradhan, S., Srivastva, A., & Aggarwal, L. (2010). Automated medical image segmentation techniques. *Journal of Medical Physics*, *35*(1), 3. https://doi.org/10.4103/0971-6203.58777

Sharma, S., Ram, S., Kumar, S., Saxena, R., & Kumar, R. (2015). Expert Systems with Applications Identity verification using shape and geometry of human hands. *Expert Systems With Applications*, *42*(2), 821–832. https://doi.org/10.1016/j.eswa.2014.08.052

Simmons, K. P., & Istook, C. L. (2003). *Body measurement techniques*. *Journal of Fashion Marketing and Management: An International Journal* (Vol. 7). https://doi.org/10.1108/13612020310484852

Stancic, I., Supuk, T., & Cecic, M. (2009). Computer vision system for human anthropometric parameters estimation. *WSEAS Transactions on Systems*, *8*(3), 430–439.

Stockslager, M. A., Capocasale, C. M., Holst, G. L., Simon, M. D., Li, Y., McGruder, D. J., … Forest, C. R. (2016). Optical method for automated measurement of glass micropipette tip geometry. *Precision Engineering*, *46*, 88–95. https://doi.org/10.1016/j.precisioneng.2016.04.003

Stomfai, S., Ahrens, W., Bammann, K., Kovács, Mårild, S., Michels, N., … Molnár, D. (2011). Intra- And inter-observer reliability in anthropometric measurements in children. *International Journal of Obesity*, *35*, S45–S51. https://doi.org/10.1038/ijo.2011.34

Tabbone, S., De, D., Cnrs, C., & Lorraine, I. (n.d.). Edge Detection Techniques - An Overview, 1–41.

Takami, O., Irie, N., Kang, C., Ishimatsu, T., & Ochiai, T. (1996). Computer interface to use head movement for handicapped people. *Proc. of the IEEE TENCON on Digital Signal Processing Applications*, 468–472. https://doi.org/10.1109/TENCON.1996.608861

Uhm, T., Park, H., & Park, J. Il. (2015). Fully vision-based automatic human body measurement system for apparel application. *Measurement: Journal of the International Measurement Confederation*, *61*, 169–179. https://doi.org/10.1016/j.measurement.2014.10.044

Vásquez, A., & Perdereau, V. (2017). Proprioceptive shape signatures for object manipulation and recognition purposes in a robotic hand. *Robotics and Autonomous Systems*, *98*, 135–146. https://doi.org/10.1016/j.robot.2017.06.001

Weinberg, S. M., Scott, N. M., Neiswanger, K., Brandon, C. A., & Marazita, M. L. (2004). Digital three-dimensional photogrammetry: Evaluation of anthropometric precision and accuracy using a Genex 3D camera system. *Cleft Palate-Craniofacial Journal*, *41*(5), 507–518. https://doi.org/10.1597/03-066.1

Wibowo, R. K. K., & Soni, P. (2014). Anthropometry and Agricultural Hand Tool Design for Javanese and Madurese Farmers in East Java, Indonesia. *APCBEE Procedia*, *8*(Caas 2013), 119–124. https://doi.org/10.1016/j.apcbee.2014.03.012

Yu, A., Yick, K. L., Ng, S. P., & Yip, J. (2013). 2D and 3D anatomical analyses of hand dimensions for custom-made gloves. *Applied Ergonomics*, *44*(3), 381–392. https://doi.org/10.1016/j.apergo.2012.10.001

Zatočilová, A., Paloušek, D., & Brandejs, J. (2016). Image-based measurement of the dimensions and of the axis straightness of hot forgings. *Measurement: Journal of the International Measurement Confederation*, *94*, 254–264. https://doi.org/10.1016/j.measurement.2016.07.066

Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, *29*(8), 1335–1346. https://doi.org/10.1016/0031-3203(95)00169-7

Zhu, L., Khoramshahi, E., Turppa, T., & Salminen, E. (2015). Traffic queue length measurement by using combined methods of Photogrammetry and digital image processIng, (1995), 4–7. https://doi.org/10.1109/JURSE.2015.7120531

APPENDIX A

JOURNAL LIST

Number	Title
1	An Anthropometric Survey of Korean Hand and Hand Shape Types
2	Anthropometric Characteristics of The Hand Based on Laterality and Sex Among Jordanian
3	Anthropometry and Agricultural Hand Tool Design for Javanese And Madurese Farmers in East Java, Indonesia
4	Determination of Hand and Palm Area as A Ratio of Body Surface Area in Indian Population
5	Hand Anthropometry Survey for The Jordanian Population
6	Hand Measurement Data from Human Babies at Birth, from 26 To 41 Weeks Estimated Gestational Age
7	Intra- and Inter- Observer Reliability in Anthropometric Measurements in Children
8	Sex Determination from Hand Dimensions and Index or Ring Finger Length Ratio in North Saudi Population, Medico-Lega View
9	The Effects of Tool Handle Shape on Hand Performance, Usability and Discomfort Using Masons' Trowels
10	2D and 3D Anatomical Analyses of Hand Dimensions for Custom-Made Gloves
11	Relative Measurement Theory- The Unification of Experimental and Theoretical Measurements
12	The Role of Human Fatigue in the Uncertainty of Measurement
13	3D Hand Anthropometry of Korean Teenagers and Comparison with Manual Method
14	Anthropometric Body Measurements Based on Multi-View Stereo Image Reconstruction

Number	Title
15	Body Measurement Techniques-Comparing 3D Body-Scanning and Anthropometric Methods for Apparel Applications
16	Comparison of Human Body Sizing Measurement Data by Using Manual And 3D Scanning Measuring Techniques
17	Digital 3 -Dimensional Photogrammetry- Evaluation of Anthropometric Precision and Accuracy Using a Genex 3D Camera System
18	Hand Anthropometry of Colombian Caribbean College Students Using Software Based Method
19	Human-Sized Anthropomorphic Robot Hand with Detachable Mechanism at The Wrist
20	Measurement of Torsion from Multitemporal Images of The Eye Using Digital Signal
21	Vision-Based Human Grasp Reconstruction Inspired by Hand Posture Synergies
22	Automated Vehicle Classification with Image Processing and Computational Intelligence
23	Computer Interface to Use Head Movement for Handicapped People
24	Fabric Drape Measurement- A Modified Method in Digital Image Processing
25	Hand-Segmentation from Depth Image Using Anthropometric Approach
26	Production System Geometry Assurance Using 3D Imaging
27	Proprioceptive Shape Signatures for Object Manipulation and Recognition Purposes in A Robotic Hand
28	Unsupervised Construction of Human Body Models
29	A Review on Image Segmentation Techniques
30	A Survey on Evaluation Methods for Image Segmentation
31	An Overview on Image Processing Techniques
32	Automated Medical Image Segmentation Techniques
33	Efficient Graph Based Image Segmentation
34	Evaluation of Some Engineering Properties of Cucumber (Cucumis Sativus L.) Seeds and Kernels Based On Image Processing

Number	Title
35	Identity Verification Using Shape and Geometry of Human Hands
36	Image Segmentation Techniques
37	Parametric Human Body Shape Modeling Framework for Human_2012_Computer Aided
38	Real-Time Segmentation of Strawberry Flesh and Calyx from Images of Singulated Strawberries During Postharvest Processing
39	Review of MR Image Segmentation Techniques Using Pattern Recognition
40	Selecting Suitable Image Dimensions for Scanning Probe Microscopy
41	Traffic Queue Length Measurement by Using Combined Methods of Photogrammetry and Digital Image
42	Anthropometric Measurements from Photographic Images
43	Color Image Segmentation Advances and Prospects
44	Computer Vision System for Human Anthropometric Parameters Estimation
45	Fully Vision Based Automatic Human Body Measurement for Apparel Application
46	Geometry Based Mass Grading of Mango Fruits Using Image Processing
47	Interactive Image Segmentation by Maximal Similarity Based Region Merging
48	On the Color Image Segmentation Algorithm Based on the Thresholding and The Fuzzy C-Means Techniques
49	Vegetation Measurement Using Image Processing Method
50	A survey of edge detection techniques
51	Assessment of Quality of Rice Grain Using Optical and Image Processing Technique
52	Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing
53	Comprehensive Measurement System for Screw Thread Parameter Based on Machine Vision
54	Computer Vision Algorithms for Measurement and Inspection of External Screw Threads

Number	Title
55	Edge Detection in Range Images Based on Scan Line Approximation
56	Edge Detection Techniques- An Overview
57	Edge Flow- A Framework of Boundary Detection and Image Segmentation
58	Edge Detection Techniques for Image Segmentation-A Survey of Soft Computing Approaches
59	Image Segmentation Using Edge Detection
60	Image-Based Measurement of Dimensions and of The Axis Straightness of Hot Forgings
61	Methods of Image Edge Detection- A Review
62	Optical Method for Automated Measurement of Glass Micropipette Tip Geometry
63	Precise Evaluation of Anthropometric 2D Software Processing of Hand in Comparison with Direct Method
64	Scale Space and Edge Detection Using Anisotropic Diffusion
65	Study and Comparison of Various Image Edge Detection Techniques
66	User Authorization Based on Hand Geometry Without Special Equipment
67	3D Whole Body Scanners Revisited
68	Analyzing Microtomography Data with Python and The Scikit-Image Library
69	Error Detection in 3-Dimensional Surface Anthropometric Data
70	Teaching Image Processing in Engineering Using Python

APPENDIX B

SUMMARIZED OF THE JOURNALS REVIEW

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
1	Soo-chan Jee, Myung Hwan Yun	2015	To determine hand shapes and categorized those Koreans' hand.	Digital calipers, digital scales and a stadiometer to measure hand length, breadth, thickness and fingers and to measure body weight and stature.	Anthropometric data to recognize Koreans' hand shape.	Manual anthropometric measurement
2	Yunis A.A. Mohammad	2005	To study the hand dimension of left and right-handed individuals between the gender and different genders.	A wooden cone and a sliding caliper to measure internal grip diameter of hand and hand dimensions respectively.	Hand Anthropometric characteristics among Jordanian.	Manual anthropometric measurement
3	Robertoes Koekoeh K. Wibowo, Peeyush Soni	2014	To reveal the anthropometric measurements of Javanese farmers for agricultural hand tool design.	Digital caliper is used for hand measurement, Martin anthropometer is used for standing and sitting posture while a squeeze dynamometer is used to measure the farmer hand's power for data collection..	Anthropometric data for design tools among Javanese farmers.	Manual anthropometric measurement
4	Pawan Agarwal and Sashikant Sahu	2010	To determine the hand and palm area as a ratio of body surface area in Indian population.	Hand and palm surface area was calculated by using a hand tracing on plain paper.	Body surface area burn estimation.	Manual anthropometric measurement
5	Nabeel Mandahawi, Sheik Imrhan, Salman Al-Shobaki, B. Sarder	2008	To determine the hand anthropometry for Jordanian population	An electronic digital caliper with an accuracy of 0.01mm is used to measure right hand dimensions.	Hand Anthropometry survey on Jordanian population.	Manual anthropometric measurement

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
6	Emmanuelle Honoré, Thameur Rakza, Philippe Deruelle	2016	To collect hand measurement data for human babies from 26 to 41 estimated gestational age.	A medical caliper is used to measure width of the index finger, thumb and hand and length of the hand (maximal length), middle finger and palm of the hand and ray of the thumb.	Morphometric data collection	Manual anthropometric measurement
7	S Stomfai, W Ahrens, K Bammann, Ě Kovács, S Mårild, N Michels, LA Moreno, H Pohlabeln, A Siani, M Tornaritis, T Veidebaum, D Molnár	2011	To describe the standardization and reliability of anthropometric measurement.	Having central training and local training to carry out the intra- and inter-observer reliability test for at least 20 children in each center.	Intra- and inter-observer reliability in anthropometry measurements	Manual anthropometric measurement
8	Mahrous AbdelBasset Ibrahim, Athar Mohamed Khalifa, Abeer Mohamed Hagras, Naif Ibrahim Alwakid		To investigate a reliable method to determine gender through hand dimensions in Saudi population.	Measurement for hand length, breadth, hand index in cm using a Sliding caliper and index and ring finger ratio was estimated using calculation.	Sex determination using anthropometric data	Manual anthropometric measurement
9	Iman Dianat, Moein Nedaei, Mohammad Ali Mostashar Nezami	2014	To evaluate the design of hand tool handles.	Using a prototype mason's trowel to study the pressure distribution on the contact area between hand and handle and depends on the hand size to determine the degree of discomfort.	Hand grip comfort	Manual anthropometric measurement
10	Ken Krechmer	2018	To propose a formal development of an experimental measurement.	Using quantum mechanics (QM) theory and experiment for measurement on calibration and resolution.	Calibration function on measuring.	Manual anthropometric measurement
11	F. Fruggiero, M.Fera, A. Lambiase, P.Maresca, J.Caja	2017	Propose a solution model for technical and human error in measurement.	Human errors follow a System Dynamics approach	Uncertainty of manual measurement	Manual anthropometric measurement
12	A Yu, K.L. Yick, S.P. Ng, J.Yip	2012	To evaluate 2D and 3D scanning and compare with manual measurement.	Plaster hand models for subjects were created and the dimensions were compared with the real hands using measuring tape and caliper.	Hand dimensions for custom-made gloves	Vision anthropometric measurement

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
13	Se Jin Park, Seung Nam Min, Heeran Lee, Murali Subramaniyam, Sang Jae Ahn	2015	To compare 3D hand anthropometry with manual measurement for Korean Teenager.	For manual measurement, spreading caliper and martin anthropometer were used to measure hand anthropometry. For 3D measurement, a NEXHAND H-100, Knitech scanner with accuracy of 0.5mm were used to measure hand anthropometry.	Gloves manufactures and to develop national anthropometric database.	Vision anthropometric measurement
14	Zhaoxin Li, Wenyan Jia, Zhi Hong Mao, Jie Li, Hsin Chen Chen, Wangmeng Zuo, Kuanquan Wang, Mingui Sun	2013	To construct 3D body anthropometric measurements.	Using a digital camera and a simple rotating disk to conduct self-measurements of anthropometric metrics.	Anthropometric parameters in households and clinics.	Vision anthropometric measurement
15	Karla P. Simmons, Cynthia L. Istook	2003	To compare body-scanning measurement extraction methods and terminology with traditional anthropometric methods.	Using three different scanners, $[TC]^2$, Cyberware and SYMCAD to do scanning and compare the result to identify which scanner has the capability to produce most measures.	Apparel industry	Vision anthropometric measurement
16	Jing Qi, Xin Zhang, Boan Ying, Feifei Lv	2011	To study the deviation between manual measurement and 3D interactive measurement.	17 typical measurement items were used for manual measurement and ScanWorX4.0.1 software with Vitus Smart 3D human body scanner were used for 3D interactive measurement.	Guideline in using scanning equipment and software	Vision anthropometric measurement
17	Seth M. Weinberg, Nicole M. Scott, Katherine Neiswanger, Carla A. Brandon, Mary L. Marazita	2003	To determine the precision and accuracy of facial anthropometric measurements using digital 3D photogrammetry.	Two independent observes measure craniofacial using calipers and 3D photos without facial landmarks labeled. Then, the results were compared statistically including mean comparison.	Medical and craniofacial research designs.	Vision anthropometric measurement
18	Manlio Massiris, R. Peñja-Baena, Ó. Oviedo-Trespalacios, Marla Maestre-Meyer	2015	To reduce error in some body measurements using software based measurement.	Measure using a portable computer, a camera and designed software which can generate regressions to reduce error in measurements.	Ergonomic design of tools, workstations and assistant for handicapped people.	Vision anthropometric measurement

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
19	Yuichi Kurita, Yasuhiro Ono, Atsutoshi Ikeda, Tsukasa Ogasawara	2010	To develop a detachable mechanism of human-sized multi-fingered robot with at the wrist.	Vision-based slip margin estimation for tactile sensation and manipulation for grip force control.	Tendon-driven mechanism for human driven actuator part.	Vision anthropometric measurement
20	J. Anthony parker, Robert V. Kenyon, Lawrence R. Young	1985	To develop a digital signal processing technique to measure ocular torsion of the eye based on the image.	The 512 x 512 pixels picture is proceeded with translation and rotation algorithms to define the accuracy using standard deviation of error.	Eye ocular torsion measurement in medical field.	Vision anthropometric measurement
21	Ritwik Chattaraj, Siladitya Khan, Deepon Ghose Roy, Bikash Bepari, Subhasis Bhaumik	2017	To develop a marker-based hand pose-tracking solution using a Kinect depth-capture device.	Using RGB image for extraction and centroid identification of fingertip to computerize a 3D human hand position for animated human hand implementation.	Re-instating the notion of human hand performance.	Vision anthropometric measurement
22	Selim S. Sarikan, A. Murat Ozbayoglu, Oguzhan Zilci	2017	Vehicles classification in dedicated lanes using image processing and machine learning techniques.	Implemented vehicle classification software based on OpenCV, Weka and Java and result testing using two machine learning methods.	Time warping for speed variations in traffic.	3D gesture and image processing technique
23	Osamu Takami, Naoki Irie, Chulung Kang, Takakazu Ishimatsu, Tsumoru Ochiai	1996	To develop a computer interface device based on the movement of head and breathing of handicapped people.	3D position and posture detects LED lights on glasses to activate the algorithm to move the computer cursor.	Contribution in Movement for handicapped people.	3D gesture and image processing technique
24	Narahari Kenkare, Traci May	2005	To measure fabric drape measurement using digital image processing.	To digitally capturing image and using image processing software to process it.	Fabric drape evaluation	3D gesture and image processing technique
25	Rayi Yanu Tara, Paulus Insap Santosa, Teguh Bharata Adji	2012	To study human posture and human dimension of the anthropometry based on image thresholding technique using depth image method.	From depth image to image thresholding followed by a few simple steps to segmented image for hand segmentation.	Development of natural interface between human and robot using hand gesture.	3D gesture and image processing technique

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
26	Jonatan Berglund, Erik Lindskog, Johan Vallhagen, Zhiping Wang, Cecilia Berlin, Björn Johansson	2016	To evaluate 3D imaging and algorithm to assess the validity of virtual production system models.	3D laser scanning for covering comprehensive spatial data collection of production environments followed by a spatial data process to remove noise and lastly comparison between CAD model and feature.	Geometry production system assurance	3D gesture and image processing technique
27	Alex Vásquez, Véronique Perdereau	2017	To generate a shape proprioceptive signature of robotic hand for object recognition and identification.	Joint angles of a robotic hand are generated to perform object shape identification grasping based on the dynamic time warping algorithm which measure the signatures on the image taken.	Object manipulation and recognition robotic hand	3D gesture and image processing technique
28	Thomas Walther, Rolf P. Würtz	2017	To generate automatic behavior understanding for learned model based on human motion.	Implemented an artificial vision system for object properties persistent to learn the rule of collinearity from common fate and perform model adaptation and enhancement loops.	Unsupervised motion learning body model	3D gesture and image processing technique
29	Nikhil R. Pal, Sankar K. Pal	1993	To study the image range of segmentation and magnetic resonance images.	Image processing and gray level histogram for segmentation.	Magnetic resonance image (MRI)	Grayscale
30	Y.J. Zhang	1995	To study different methods used for segmentation evaluation.	Analytical, empirical goodness and empirical discrepancy groups.	Image segmentation characteristics	Grayscale
31	B. Chitradevi, p. Srimathi	2014	To review the image processing technique.	Image processing, image acquisition, image enhancement, image segmentation, feature extraction, image classification.	Human interpretation, image data processing, machine perception transmission and representation.	Grayscale
32	Neeraj Sharma, Lalit M. Aggarwal	2010	To discuss the problems encountered in segmentation of CT and MR images.	Image segmentation on gray level/histogram features for edge segmentation and region segmentation. Use textural features for statistical approach, syntactic/structural approach and spectral approach.	Magnetic Resonance (MR) and Computed Topography (CT)	Grayscale

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
33	Pedro F. Felzenszwalb, Daniel P. Huttenlocher	2003	To suggest method in solving problem of segmenting an image into regions.	Develop an efficient segmentation algorithm based on measurement boundary between two regions using a graph-based representation of the image and the algorithm is using two different kinds of local neighborhoods in constructing the graph.	Image segmentation application	Grayscale
34	Amir Hossein Mirzabe, Masoud Barati Kakolaki, Behnam Abouali, Rasoul Sadin	2016	To investigate the effect of moisture content on gravimetrical and frictional properties of the seeds and kernels of cucumber based on digital image processing method.	Digital image processing used for measuring parameter dimension and then followed by gravimetric and frictional approaches.	Cucumber industry and agriculture	Grayscale
35	Shefali Sharma, Shiv Ram Dubey, Satish Kumar Singh, Rajiv Saxena, Rajat Kumar Singh	2014	To identify personal hand shape and hand geometry verification using image segmentation.	The algorithm involves image acquisition, image preprocessing to convert the image to gray scale, hand orientation registration and reference point extraction.	Multimodal biometric system	Grayscale
36	Robert M. Haralick, Linda G. Shapiro	1984	To define every image segmentation technique.	Image segmentation techniques are measurement space guided spatial clustering, single linkage region growing schemes, hybrid and centroid linkage region growing schemes, spatial clustering schemes and split and merge schemes.	Image segmentation technique	Grayscale
37	Seung-Yeob Baek, Kunwoo Lee	2010	To study the integration of novel parametric human body shape modeling framework into product design applications.	3D digital image processing from direct model creation, template model based scaling, image based reconstruction and statistics based model synthesis.	Human centered product design	Grayscale
38	A. Durand-Petiteville, S. Vougioukas, D.C. Slaughter	2017	To extract the flesh and calyx area from strawberry image using image processing.	Image acquisition, RGB image to grayscale image conversion to thresholding to extracts binary flesh and binary calyx images.	Food and agriculture	Grayscale

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
39	J.C. Bezdek, L.O. Hall, L.P. Clark	1993	To investigate pattern recognition used in MR image segmentation.	Pattern recognition starts with fuzzy model approach, the mixture model and lastly computational neural networks (CNNs).	Medical Resonance Imaging (MRI)	Grayscale
40	James Bowen, David Cheneler	2017	To investigate the correct image dimensions to scan probe microscopy.	To capture the sample on the steel specimen disks using nanotechnology and cyanoacrylate adhesive prior to measurement.	Topographical information and surface metrology	Grayscale
41	Lingli Zhu, Ehsan Khoramshahi, Tuomas Turppa, Eero Salminen	2015	To measure a traffic queue length on real time by using a combination of photogrammetry and image processing methods.	One camera is fixed under a bridge to capture digital image to enable image processing to extract the interior orientation and exterior orientation to control the time for the traffic.	Time variation to control traffic queue length.	Grayscale
42	Patrick Chi- Yuen Hung, Channa P. Witana, Ravindra S. Goonetilleke	2004	To obtain anthropometric data from photographic images.	Canon IXUS color digital camera was used to capture the front view, back view and side view of a person and using manual digitization of landmarks to extract the2D measurements for circumference and linear measurements.	Anthropometric measurements	Color
43	H.D. Cheng, X.H. Jiang, Y.Sun, Jingli Wang	2000	To review available color segmentation techniques.	Using color segmentation to proceed histogram thresholding, characteristic feature clustering, edge detection, region-based methods, fuzzy techniques and neural networks.	Color image segmentation	Color
44	Ivo Stancic, Tamara Supuk, Mojmil Cecic	2009	To develop a computer vision system to measure human anthropometric parameters.	Using color subject's image with aid of algorithm to create 3D model of the human from 2D image and ellipses with basic building objects for measurement.	Human anthropometric parameters	Color
45	Taeypung Uhm, Hanhoon park, Jong-II Park	2014	To develop vision based method to measure the dimensions of subject's body.	Background subtraction and marker recognition, frontal image processing for robust marker detection, foot image processing for manufacturing customized insole and measurement taken from the image.	Apparel industry	Color

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
46	M.A. Momin, M.T. Rahman, M.S. Sultana, C. Igathinathne, A.T.M. Ziauddin, T.E. Grift	2017	To automate the grade of mangos in terms of geometry and shape using digital image processing.	Using region based global thresholding color binarization algorithm with a combination of median filter and morphological analysis to classify the mango grades.	Food and agriculture	Color
47	Jifeng Ning, Lei Zhang, David Zhang, Chengke Wu	2009	To develop an interactive image segmentation to merge a new region.	Starts with image captured and maximal similarity based region merging using develop algorithm to progressively assign both of the non-maker background regions.	Color image segmentation	Color
48	Young Won Lim, Sang Uk Lee	1989	To present a color segmentation algorithm based on thresholding and fuzzy c-means techniques.	Scale-space filter to analyze the RGB histograms and coarse-fine concept to reduce the computational burden required for fuzzy c-means and followed by thresholding technique to carry out segmentation.	Color image segmentation	Color
49	Mürvet Kirci, Ece Olcay Günes, Yüksel Cakir	2014	To develop a digital visual assessment for measuring the vegetation cover and analyze it.	Image acquisition in RGB matrices to preprocessing it based on RGB color histogram to determine the greenness value of the image.	Food and agriculture	Color
50	Larry S. Davis	1974	To review available edge detection techniques.	The research starts with problem of edge detection identification, linear parallel edge detectors, non-linear parallel edge detectors, optimal approaches for edge detection and lastly the review on sequential edge detection.	Edge detection and segmentation	Edge and line
51	Zahida Parveen, Dr. Muhammad Anzar Alam, Hina Shakir	2017	To define the assessment of quality of rice grain using optical and image processing technique.	Extended maxima operator detects the chalky area in the rice and use image processing with algorithm to calculate the dimensions and color to classify the quality of the rice grains.	Food and agriculture	Edge and line
52	Jianping Fan, David K.Y. yau, Ahmed K. Elmagarmid, Walid G. Aref	2001	To integrate color-edge extraction and seeded region growing.	It involves edge detection, face detection, image segmentation and seeded region growing procedure.	Edge detection and segmentation	Edge and line

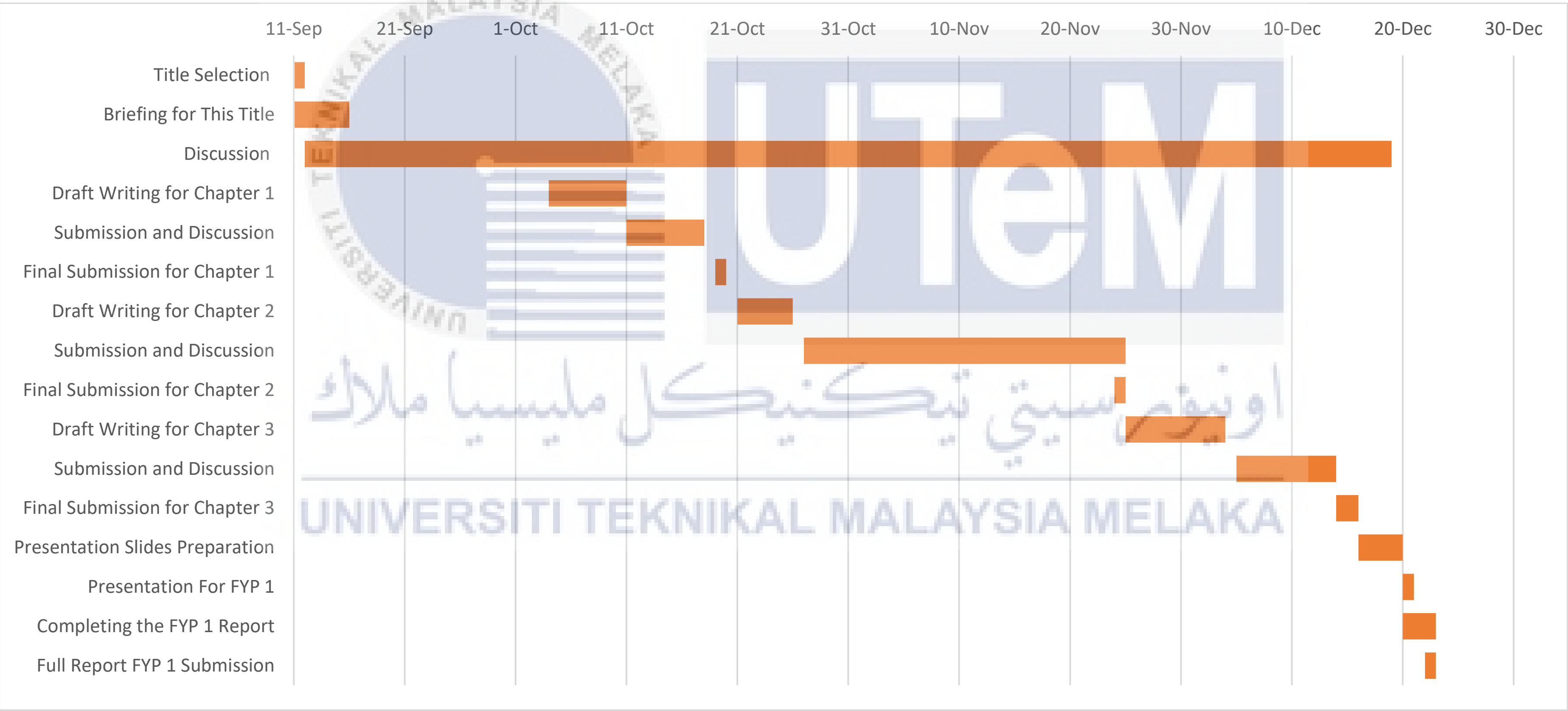
No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
53	Jing Min	2015	To measure the tooth angle and geometrical parameter using machine vision.	It includes image smoothing, edge detection, binary image producing and contour hunting.	Quality assurance in screw thread measurement	Edge and line
54	E.S. Gadelmawla	2016	To develop a vision system for automatic measure and inspect the external screw threads.	After system set up, it is followed by image processing algorithms, computer vision algorithms, measuring and inspecting the screw threads.	Quality assurance in screw thread parameter measurement	Edge and line
55	Xiaoyi Jiang, Horst Bunke	1998	To present a novel edge detection algorithm using scan line approximation technique.	Edge detection using simple contour closure technique to segment image into regions.	Edge and line detection and segmentation	Edge and line
56	Djemel Ziou, Salvatore Tabbone	1996	To present an overview of research in edge detection.	Edge definition, properties of detectors, the methodology of edge detection, the mutual influence between edges and detectors and existing edge detectors with implementation.	Edge and line detection and segmentation	Edge and line
57	W.Y, Ma, B.S. Manjunath	1997	To identify the direction of color change and texture at each image location and construct an edge flow vector.	Edge detection, texture segmentation, "edge flow" identification, intensity edge flow, texture edge flow, edge flow based on phase and then edge flow integration and flow propagation and boundary detection, lastly the boundary connection and region merging.	Boundary detection and segmentation	Edge and line
58	N. Senthilkumaran, R. Rajesh	2014	To survey the edge detection theory for image segmentation using soft computing approach.	It involves image segmentation, edge detection, fuzzy logic, genetic algorithm and neural network.	Soft computing on edge detection and segmentation	Edge and line
59	Salem Saleh Al-amri, N.V. Kalyankar, Dr Khamitkar S.D.	2010	To identify the best edge detection technique of the satellite image.	Seven edge detection techniques are Sobel operator technique, Prewitt technique, Kiresh technique, Laplacian technique, Canny technique, Roberts technique and Edge Maximization technique (EMT).	Edge detection, discontinuity detection/types of discontinuity detection.	Edge and line

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
60	Aneta Zatočilová, David Paloušek, Jan Brandejs	2016	To measure the dimensions and axis straightness of hot forgings using vision based measurement with edge detection method.	Two high resolution single-lens reflex camera and algorithms to extract the 3D model construction for measurement.	Hot forgings measurement	Edge and line
61	Dharampal, Vikram Mutneja	2015	To introduce the available edge detection methods.	It describes general edge detection and edge detection method based on Laplacian.	Edge and line detection techniques	Edge and line
62	Max A. Stockslager, Christopher M. Capocasale, Gregory L. Holst, Michael D. Simon, Yuanda Li, Dustin J. McGruder, Erin B. Rousseau, William A. Stoy, Todd Sulchek, Craig R. Forest	2016	To develop a digital optical method to automated measure the glass micropipette tip geometry.	It starts from acquire image and compute focus measure F(i), microscope and micropipette fixture, pipette tip-finding algorithm, autofocus algorithm, tip outer diameter and cone angle measurement algorithm, cone angle correction and system characterization.	Glass micropipette tip measurement	Edge and line
63	Ehsanollah Habibi, Shiva Soury, Akbar Hasan Zadeh	2013	To test the efficiency of 2D image processing software in hand photo anthropometry.	Measure both using digital caliper for manual measurement and 2D photo anthropometry and then edge segmentation for digital measurement.	Human anthropometry data	Edge and line
64	Pietro Perona, Jitendra Malik	1990	To investigate anisotropic diffusion using scale-space and edge detection.	Anisotropic diffusion to determine the maximum principle, edge enhancement.	Edge and line detection techniques	Edge and line
65	Raman Maini, Dr. Himanshu Aggarwal	2002	To determine the edge detection using gradient based and Laplacian based detection.	Gradient based and Laplacian based edge detection.	Edge and line detection techniques	Edge and line
66	Marek Klonowski, Marcin Plata, Piotr Syga	2017	To develop a hand geometry user identification algorithm.	2D image development followed by edge and line detection and segmentation to identify the hand geometry.	Bio-identification and biometric	Edge and line

No.	Author	Year	Objective	Parameters, Methods and Tools	Application	Classification
67	H.A.M. Daanen, F.B. Ter Haar	2013	To review 3D scanning system typically for laser line systems. structured light systems, multi-view camera system, millimeter waves and a review on software.	A general introduction on laser line systems. structured light systems, multi-view camera system, millimeter waves and software introduction on easy optimization and repair of the scanned images, database to browse the information with dedicated software.	Garment industry	Computer vision method based on Python
68	Emmanuelle Gouillart, Juan Nunez-Iglesias, Stéfan van der Walt	2016	To analyze microtomography data using Python and scikit-image library.	An overview on the scikit-image pattern and short snippets of code developed using Python. In image processing, Python is used to generate NumPy arrays and further description on the Python ecosystem.	Scikit-image and Python on developing image processing algorithms.	Computer vision method based on Python
69	Jinwoo Park, Yunja Nam, Eunkyung Lee, Sunmi Park	2008	To analyze error detection in 3D surface anthropometric data.	3D body scanning, data checking on logical and graphical checking.	3D surface anthropometry data	Computer vision method based on Python
70	Andrés Fernando Jiménez López, Marla Carolina Prieto Pelayo, Ángela Ramírez Forero	2016	To describe manipulation of image processing using Python programming language.	Python language using OpenCV, numpy, pyqt, sys, os and matplotlib.	Python programming language	Computer vision method based on Python

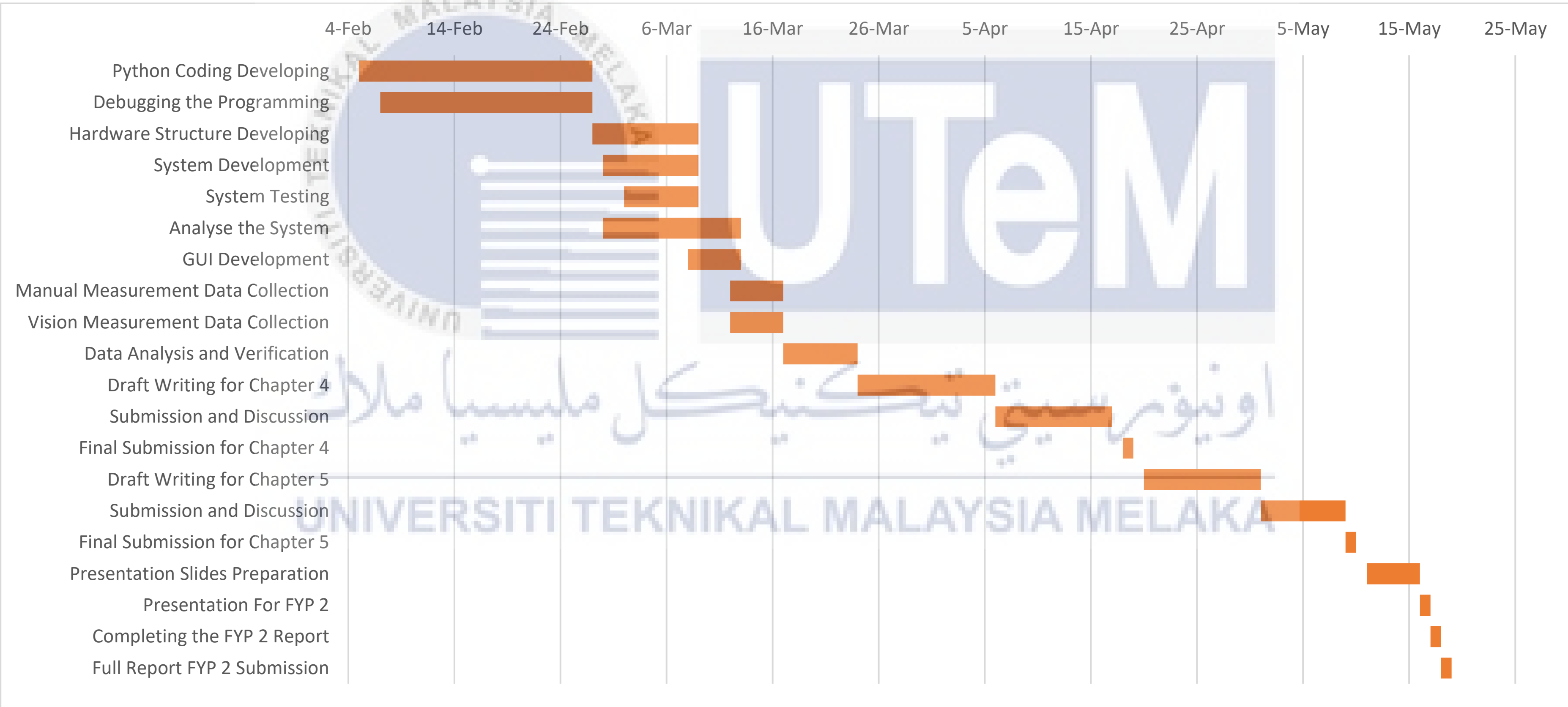
APPENDIX C

GANTT CHART FOR FYP I



APPENDIX D

GANTT CHART FOR FYP II



APPENDIX E

SOURCE CODE

```
from picamera.array import PiRGBArray
from picamera import PiCamera
from scipy.spatial import distance as dist
from sklearn.metrics import pairwise
from imutils import contours
import time
import imutils
import cv2
import numpy as np
import os
import sys
import math

PY3 = sys.version_info[0]==3
if PY3:
    xrange=range

#declare PiCamera as camera
camera=PiCamera()
#set camera resolution
camera.resolution =(650,700)
#construct an absolute path to the directory of this file located
path=r"/home/pi/Desktop/Project"

---------------------------------------------CAMERA_SETUP --------------------------------------
---
```

```
def camera_setup():

    #boolen expression
    done=False

    while not done:
        #declare PIRGBArray
        rawCapture = PiRGBArray(camera)
        #include brightness
        rawCapture.brightness=70
        #allow the camera to warm up
        time.sleep(0.1)
        #ensure the image format is "bgr"
        camera.capture(rawCapture, format='bgr')
        #clarify array
        image = rawCapture.array
        #convert RGB to BGR because openCV is using BGR and picamera is RGB
        image = cv2.cvtColor(image,cv2.COLOR_RGB2BGR)
        #display the image on screen and wait for a keypress
        cv2.imshow("Image", image)
        #wait for key press to continue
        cv2.waitKey(0)
        # clear the stream in preparation for the next frame
        rawCapture.truncate(0)
        #user input for the camera image setting
        choice=input("The camera setup satisfy? (y/n)\n").strip()
        #if the answer is no,"n"
        if choice =='n':
            #camera setup testing will be repeated
            print("Repeat camera setup testing")
        else:
            break
    done=True
```

--IMAGE_CAPTURE--

```
def image_capture():

   #allow the camera to warm up
   time.sleep(0.1)
   #declare PiRGBArray
   rawCapture = PiRGBArray(camera)
   #include brightness
   rawCapture.brightness=70
   #ensure the image format is BGR
   camera.capture(rawCapture, format="bgr")
   #clarify array
   image = rawCapture.array
   #camera warm up time
   time.sleep(0.1)
   #convert RGB to BGR because openCV is BGR and picamera is RGB
   image = cv2.cvtColor(image,cv2.COLOR_RGB2BGR)
   #display image
   cv2.imshow("Image", image)
   #wait for key press to continue
   cv2.waitKey(0)
   #inform user the image is taken
   print("Your palm is captured!")
   #keep the image in this path
   old_image=cv2.imwrite(os.path.join(path,'Subject.png'),image)
   #wait for key press to continue
   cv2.waitKey(0)
   #clear the stream in preparation for the next frame
   rawCapture.truncate(0)
```

------------------------------------LABEL_HAND_CONTOUR---------------------------------------

```
def get_hand_contour(image):

    #grabbed the current image
    image= cv2.imread("//home//pi//Desktop//Project//Subject.png")
    #convert it to grayscale
    gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    #apply gaussianBlur
    gaussianBlur = cv2.GaussianBlur(gray, (7, 7), 0)
    #apply threshold to smooth the image
    threshold = cv2.threshold(gaussianBlur, 45, 255, cv2.THRESH_BINARY)[1]
    #apply dilation to remove background noise
    dilate = cv2.dilate(threshold, None, iterations=2)
    #apply erosion to remove foreground noise
    erode = cv2.erode(dilate, None, iterations=2)
    #find contours for the hand
    cnts = cv2.findContours(erode.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    #checking Python Version to be compatible with cnts function
    cnts = cnts[0] if imutils.is_cv2() else cnts[1]
    #based on contour area to get the minimum contour
    c = min(cnts, key=cv2.contourArea)
    #to ensure the it is closed curve contour
    perimeter = cv2.arcLength(c, True)
    #to obtain contour approximation for a near perfect contour
    epsilon = 0.0002*cv2.arcLength(c,True)
    approx = cv2.approxPolyDP(c, epsilon, True)
    #draw the hand
    cv2.drawContours(image, [approx], -1, (0, 0, 255), 2)
    #display image
    cv2.imshow("image",image)
```

```
    #wait for key press to continue
    cv2.waitKey(0)
    #return erode and approx to the working procedure
    return(erode,approx)
```

-----------------------COUNT_FINGERS_FOR_POINTS_AT_FINGERTIPS---------------------

```
def count(erode,approx):
    #find convex hull of the hand in the image
    hull = cv2.convexHull(approx)
    #find the most extreme points in the convex hull
    extLeft = tuple(approx[approx[:, :, 0].argmin()][0])
    extRight = tuple(approx[approx[:, :, 0].argmax()][0])
    extTop = tuple(approx[approx[:, :, 1].argmin()][0])
    extBot = tuple(approx[approx[:, :, 1].argmax()][0])
    #find the center of the hand palm
    cX = int((extLeft[0] + extRight[0]) / 2)
    cY = int((extTop[1] + extBot[1]) / 2)
    centerMass=(cX,cY)
    #find the maximum euclidean distance between the center of the palm and
    #extreme points of the convex hull
    distance = pairwise.euclidean_distances([(cX, cY)],
    Y=[extLeft, extRight, extTop, extBot])[0]
    maximum_distance = distance[distance.argmax()]
    #calculate the radius of the circle with 60% of the max euclidean distance obtained
    radius = int(0.6 * maximum_distance)
    #find the circumference of the circle
    circumference = (2 * np.pi * radius)
    #take the circular region of interest which has the palm and the fingers
    circular_roi = np.zeros(erode.shape[:2], dtype="uint8")
    #draw the circular ROI
    cv2.circle(circular_roi, (cX, cY), radius, 255, 1)
```

```
#take bit-wise AND between erode hand using the circular ROI as the mask
#which gives the cuts obtained using mask on the eroded hand image
circular_roi = cv2.bitwise_and(erode, erode, mask=circular_roi)
#compute the contours in the circular ROI
_,cnts,_ = cv2.findContours(circular_roi.copy(), cv2.RETR_EXTERNAL,
                        cv2.CHAIN_APPROX_NONE)

#below are the operations used to display the drawing (optional to open/close them)
##cv2.circle(image, (cX, cY), radius, (0,255,255), 1)
##cv2.circle(image, (cX, cY), 6, (0,255,255), -1)
##cv2.circle(image, extLeft, 6, (0, 0, 255), -1)
##cv2.circle(image, extRight, 6, (0, 255, 0), -1)
##cv2.circle(image, extTop, 6, (255, 0, 0), -1)
##cv2.circle(image, extBot, 6, (255, 255, 0), -1)
#display image
cv2.imshow("image",erode)
#compute the bounding box of the contour
(x,y,w,h) = cv2.boundingRect(c)
cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,0),2)
#increment the count of fingers only if
#1.the contour region is not the wrist (bottom area)
#2.the number of points along the contour does not exceed 25% of
#the circumference of the circular ROI
count=((cY + (cY * 0.25)) > (y + h)) and ((circumference * 0.25) > c.shape[0]):
#verify the object is a hand
if count < 5 && count > 5:
   #print this statement
   print("Error! This is not a hand with 5 fingers!")
   #terminate the program
   quit()

return(approx, extTop,hull)
```

---------------------MAIN_LOOP_CALCULATE_THE_MEASUREMENT-----------------------

```
def main(approx,extTop,hull):

   #declare 'pixels per metric' for calibration variable
   pixelsPerMetric = 8.26
   #measure execution time
   start_time = time.time()
   #find convex defects, let hull2 to be False
   hull2 = cv2.convexHull(approx,returnPoints = False)
   #define defects
   defects = cv2.convexityDefects(approx,hull2)
   #declare FarDefect
   FarDefect=[]
   #use a loop to find all the points
   #s=start,e=end,far=far and d=approximate distance to the farthest point
   for i in range(defects.shape[0]):
      s,e,f,d = defects[i,0]
      start = tuple(approx[s][0])
      end = tuple(approx[e][0])
      far = tuple(approx[f][0])
      #define FarDetect = far (change np.array)
      FarDefect.append(far)
      #draw line to connect start to end points
      cv2.line(image,start,end,[255,255,255],2)
      #draw circle for far points
      cv2.circle(image,far,5,[255,0,255],-1)
   #distance from each finger defect(finger webbing) to the extreme top
   distanceBetweenDefectsToextTop = []
   #declare ptA and ptB
   (ptA,ptB) =  np.array(FarDefect[3:5])
   #get the midpoint
```

```
midpoint(ptA,ptB)=((ptA[0] + ptB[0]) * 0.5, (ptA[1] + ptB[1]) * 0.5)
#get np.array of extTop
extTop = np.array(extTop)
#use distance formula
distance = np.sqrt(np.power(midpoint[0]-extTop[0],2)+ np.power(modpoint[1]-
    extTop[1],2))
#assign distance
distanceBetweenDefectsToextTop.append(distance)
#get the distances from finger webbing to extreme top
index_finger = distanceBetweenDefectsToextTop
#compute the size of the object
Finger_length = index_finger / pixelsPerMetric
#change the unit from inch to millimetre,1 inche =25.4millimetre
Finger_length=Finger_length*25.4
#stop timing
t=time.time()-start_time
#print finger length and execution time
print("Finger length : ",Finger_length,"mm","\nExecution time : ",t,"s")

#measure execution time
start_time = time.time()
#draw a rectangle using approx contour
x,y,w,h = cv2.boundingRect(approx)
cv2.rectangle(image,(x,y),(x+w,y+h),(0,200,0),2)
#get teh height value
h,w,_ =np.shape(rectangle)
#assign height to h
height = int(h)
#get palm length
palm_length=height-index_finger
#compute the size of the object
Palm_length = palm_length / pixelsPerMetric
#change the unit from inch to millimetre,1 inche =25.4millimetre
```

```
Palm_length=Palm_length*25.4
#stop timing
t=time.time()-start_time
#print palm length and execution time
print("Palm length : ",Palm_length,"mm","\nExecution time : ",t,"s")

#measure execution time
start_time = time.time()
#define length
LENGTH = len(hull[i])
#declare the status of length
status = np.zeros((LENGTH,1))
#use loop to find close points on hull
for find_if_close(p1,p2):
    row1,row2 = p1.shape[0],p2.shape[0]
    for i in xrange(row1):
        for j in xrange(row2):
            dist = np.linalg.norm(p1[i]-p2[j])
            if abs(dist) < 50 :
                return True
            elif i==row1-1 and j==row2-1:
                return False
#use another loop for hull
for i,p1 in enumerate(hull):
    x = i
    if i != LENGTH-1:
        for j,p2 in enumerate(hull[i+1:]):
            x = x+1
            dist = find_if_close(p1,p2)
            if dist == True:
                val = min(status[i],status[x])
                status[x] = status[i] = val
            else:
```

```
            if status[x]==status[i]:
                status[x] = i+1
#declare unified
unified = []
#declare minimum
minimum = int(status.min())+1
#use a loop on minimum
for i in xrange(minimum):
    pos = np.where(status==i)[0]
#use a condition to ensure the line
if pos.size != 0:
    cont = np.vstack(hull[i] for i in pos)
    hull = cv2.convexHull(cont)
    unified.append(hull[i])
#compute the size of the object
Hand_breadth = unified / pixelsPerMetric
#change the unit from inch to millimetre,1 inche =25.4millimetre
Hand_breadth=Hand_breadth*25.4
#stop timing
t=time.time()-start_time
#print hand breadth and execution time
print("Hand breadth : ",Hand_breadth,"mm","\nExecution time : ",t,"s")

#calculate finger vs. palm(shape) index
Finger_vs_Palm = (Finger_length*100)/(Palm_length + Hand_breadth)
#stop timing
t=time.time()-start_time
#print finger vs. palm
print("Finger VS Palm (shape) index : ",Finger_vs_Palm,"mm","\nExecution time : ",t,"s")

----------------------------------------WORKING_PROCEDURE-------------------------------------
```

```
if __name__ == "__main__":

    #initialize the camera
    print("Camera setup testing")
    #call camera_setup function
    camera_setup()
    #inform user the camera is ready to use
    print("The camera is ready!")
    #request user to put her/his hand and press enter
    input("Please position well your hand palm and press enter...")
    #initialize the number of pictures taken
    n=1

    #boolen expression
    while True:
        #call image_capture function
        image_capture()
        #read image
        image=cv2.imread("//home//pi//Desktop//Project//Subject.png")
        #declare hand and remove the noise in the image
        hand=get_hand_contour(image)
        #check if hand is segmented
        if hand is not None:
            #if done segmented, unpack erode and segmented region to hand
            (erode,approx)=hand
            #draw the segmented region and display it
            cv2.drawContours(image, [approx], -1, (0, 0, 255), 2)
            #counting fingers
            fingers=count(erode,approx)
            #calculate measurement
            main(approx,extTop,hull)
            # show the image
```

```
            cv2.imshow("image",image)
        #saving a final copy
        cv2.imwrite(os.path.join(path,'Subject'+str(n)+'.png'),image)
        #clarify the path for old_image
        old_image=os.path.join(path,'Subject.png')
        #clarify the path for new_image
        new_image=os.path.join(path,'Subject'+str(n)+'.png')
        #rename the old_image to new_image
        os.rename(old_image,new_image)
        #to continue or stop the program
        choice=input("Continue for another measurement? (y/n)\n").strip()
        #choose no ("n")
        if choice== "n":
            #break the loop
            break
        #continue the loop
        else:
            #add 1 to n
            n+=1
    #camera release
    camera.stop_preview()
    #camera close
    camera.close()

#print the notisfication
print("Image file is saved.\nThank you for using.\n")
#press a random key to close the window
cv2.waitKey(0)
#close all windows
cv2.destroyAllWindows()
```