Autonomous Shape and Best Grip Determination for Upper Limb Prosthetics

Author

SAQIB ZAFAR

Regn. Number: NUST201261249MCEME35512F

Supervisor

DR. MOHSIN ISLAM TIWANA

DEPARTMENT OF MECHATRONICS ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

August, 2016

Autonomous Shape and Best Grip Determination for Upper Limb Prosthetics

Author

SAQIB ZAFAR

NUST201261249MCEME35512F

Thesis submitted in partial fulfillment of the requirements for the degree of

MS Mechatronics Engineering

Thesis Supervisor

DR. MOHSIN ISLAM TIWANA

Thesis Supervisor's Signature: _____

DEPARTMENT OF MECHATRONICS ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

August, 2016

Declaration

I certify that this research work titled *"Autonomous shape and Best Grip Determination for Upper Limb Prosthetics"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Saqib Zafar

NUST201261249MCEME35512F

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the University.

Signature of Student
Saqib Zafar
NUST201261249MCEME35512F

Signature of Supervisor
Dr. Mohsin Islam Tiwana

Copyright Statement

Acknowledgements

In the beginning, the name of ALLAH, the creator and the nurturer of the all the universes. All praises are for HIM who being REHMAAN has given everything without asking and who being RAHIM has forgiven every time. Everything which will follow in this research, every bit of word and every part of knowledge is due to HIM.

I am profusely thankful to my beloved parents and Family who gave me confident to be here and be the part of such knowledge.

I would also like to express special gratitude to my supervisor Dr. Mohsin Islam Tiwana for his help throughout my thesis. He has been a great motivator. In every moment of the research he has guided me. I have learnt too many things from him a part from this project.

I would also like to express my special thanks to Brig. Dr. Akhtar Nawaz Malik and Ms. Fatma Faruq for their guidance.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and my supervisor whose tremendous support and cooperation led me to this wonderful accomplishment

Abstract

The aim of our research is to develop a prosthetic hand that is as natural and easy to use as a person's organic extremity, without the need for invasive surgical procedures, an autonomous hand capable of determining the most suitable grasping pattern for griping an object and executing that best grasping pattern with minimum human effort. By using machine vision techniques a system is developed for fully automatic object recognition and reconstruction of 3D objects from multiple images taken from single camera embedded in the palm of the hand. We assume that the objects or scenes are rigid. A camera Matrix is associated for each image, which is parameterized by rotation, translation and focal length. For Feature matching between all images we use Speeded-Up Robust Features (SURF) and by using the RANSAC algorithm noisy matches are eliminated and find those matches that are consistent with the fundamental matrix. Objects are recognized as subsets of matching images. From 3D reconstruction we can estimate the perimeters i.e. length, width and Height of detected object and by using thresh holding function that is defined by different experimentation on human hand we execute the best grasping pattern for the griping the detected object.

Key Words: *Prosthetic Hand, 3D Reconstruction, Grasping Pattern*

Table of Contents

Declaration i

Language Correctness Certificate ii

Copyright Statement iii

Acknowledgements iv

Abstract vi

Table of Contents vii

List of Figures x

List of Tables xiii

Chapter 1: Introduction 1

1.1. Introduction 1

1.2. Non-anthropomorphic prosthetic devices 3

1.3. Anthropomorphic Prosthetic Devices 3

1.4. Problem Statement: 4

1.5. Aims & Objective of the Thesis: 6

1.6. Thesis Overview 6

Chapter 2: Advanced Myo Electric Commercial & Vision Based Upper Limb Prosthetic Hands 8

2.1 Advanced Myo Electric Commercial Hands 8

2.1.1 Bebionics Myo Electric Prosthetic Hand 8

2.1.2 i-Limb ultra 9

2.2 Vision Based Upper Limb Prosthetic Hands 10

2.2.1 The IRIS hand (2014) 10

2.2.2 IRIS Hand Electrical Architecture 12

2.2.2.1 Communication 12

2.2.2.2 Finger Actuation 13

2.2.3 Limitations of IRIS Hand 13

2.2.4 Vision-based Intelligent Prosthetic Robotic Arm (2015) 14

2.2.4.1 System Overview 15

2.2.4.2 Analysis of IRIS Hand System 16

2.2.4.3 Electrical Architecture 16

2.2.4.4 VIPeR Arm electrical Architecture 16

2.2.4.5 Object Recognition Components 18
2.2.4.6 Limitations of VIPeR Arm 19
2.2.5 Robotic Grasping of Novel Objects 19
2.2.5.1 Learning the Grasping Point 20
2.2.5.2 Grasping Point 21
2.2.5.3 Results of Novel Objects 22
2.2.5.4 Limitations 22
2.2.6 PR2 Robot 23
Chapter 3: Methodology 25
3.1 Overview 25
3.2 Flow Diagram of Our System 25
3.3 Modules 26
3.4 Image Acquisition and Noise Reduction 26
3.5 The model of imaging geometry 27
3.6 Camera Calibration 28
3.6.1 Camera Calibration with Rig 30
3.6.2 Results of Camera Calibration 31
Chapter 4: Under Actuated Drive Mechanism 31
3.7 Object Recognition Methods 32
3.7.1 Appearance-based Recognition 32
3.7.1.1 Direct Correlation Method 32
3.7.1.2 Edge Detection 33
3.7.2 Feature-based Recognition 34
3.7.2.1 SIFT Detection 34
3.7.3 Fiducial Markers 34
3.7.4 SURF Feature Extraction 36
3.7.4.1 Steps of SURF Feature Extraction and matching 36
3.7.4.2 Results of SURF Feature Extraction and matching 37
3.7.5 Comparision of SIFT & SURF Features. 38
3.8 Rejection of Outliers Features 41
3.9 Image Matching 42
- Rotation vector R 42
- Translation Vector 42
- Calibration Matrix 42

3.10 3D Reconstruction 42
Chapter 4: Object identification & Execution of best gripping features 44
4.1 Overview: 44
4.2 Object Identification: 44
4.2.1 Length of the object: 44
4.2.2 Width of the object: 45
4.2.3 Depth of the object: 45
4.3 Execution of Best Gripping Feature: 45
4.3.1 Human Hand Anthropometry 45
4.3.1.1 Length of the Human Hand 46
4.3.1.2 Breadth/Width of the Human Hand 46
Chapter 5: 3D Model Based Validation 48
5.1 Overview: 48
5.2 Steps of 3D Model Based Validation: 48
5.2.1 Results of 3D model based Validation 52
5.2.2 Result 55
Chapter 6: Experimental Results and Comparison with Actual Perimeters of the object 56
6.1 Overview: 56
6.2 Results 56
Chapter 7: Conclusion & Future Work 61
7.1 Comparative Study & Contributions 61
6.3 Conclusion: 61
7.3 Future Work 61
References 62
Appendix 65

List of Figures

Figure 1.1.1: Personalized Cosmetic Prosthetic Hand by Sophie de Oliveira Barata……1
Figure 1.1.2: Typical Body Powered Trans radial Prosthesis……2
Figure 1.2.1: Common Split-Hook Terminal Prosthetic Devices……..3
Figure 1.3.1: Anthropomorphic Prosthetic Hands i-LIMB Ultra Revolution & i-LIMB Cosmetic Cover……4
Figure 1.4.1: Pro E model Pro E model of Prosthetic hand with camera embedded in the palm of hand…….. 5
Figure 2.1.1.1: Bebionic Wrist Options (small, medium and Large)……8
Figure 2.1.2.1: i-Limb Mobile App for interfacing……..10
Figure 2.2.1.1: (a) shows the 3D print model (b) shows the CAD model of IRIS hand……..11
Figure 2.2.3.1: Different Grasps of IRIS Hand……14
Figure 2.2.4.1: Vision Based Intelligent Prosthetic Robotic Arm……...15
Figure 2.2.4.2: Hand Case Design……...15
Figure 2.2.4.4.1: Electrical Architecture of VIPeR Arm…….....17
Figure 2.2.4.4.2: Flow Diagram of VIPeR Arm……..18
Figure 2.2.4.5.1: Digital Vision Module of VIPeR Arm…….....19
Figure 2.2.5.1: Robotic Grasping of Novel Objects……....20
Figure 2.2.5.1.1: Intersection of Rays from images at Grasping point……...21
Figure 2.2.5.2.1: The red points in each image show the locations most likely to be a grasping point…….22
Figure 2.2.6.1: PR2 Robotic Hand……..24
Figure 3.1.1: Flow Diagram of our Research……..25
Figure 3.2.1: Modules of our Research……...26
Figure 3.3.1: RGB input image taken from image Acquisition Tool Box……..26
Figure 3.3.2: Conversion of RGB image to Binary Image……..27
Figure 3.3.3: Binary Image After Discarding unwanted ……....27
Figure 3.6.1.1: Steps of camera calibration with Rig……..30
Figure 3.6.2.1: Steps 01: Detection of Checker board square…….....31
Figure 3.6.2.2: Steps 02: Generation of World points……....31
Figure 3.6.2.3: Step 03: Extrinsic Perimeter Visualization ……..31

Figure 3.7.3.1: Several Types of Planar Pattern Marker Systems (Fiala, 2004)……..35

Figure 3.7.4.1.1: Steps for SURF Feature Extraction and Matching…….....36

Figure 3.7.4.1.2: Step 01: Selection of 600 SURF Feature Points……..37

Figure 3.7.4.2.2: Step 03: Feature Matching by using KNN Algorithm …….....37

Figure 3.7.5.1: Efficiency of SIFT and SURF for Feature Matching……..38

Figure 3.7.5.2: Accuracy of SIFT and SURF for Feature Extraction……..39

Figure 3.7.5.3: Graph between Selection of interest Points and processing time…….....40

Figure 3.7.5.4: Graph between Selection of interest Points and Accuracy of Correct matches.....41

Figure 3.10.1: Relationship of Two Views…….…….43

Figure 3.10.2: Result of 3D Reconstruction of object……..43

Figure 4.3.1.1.1: Human Hand Anthropometry……...46

Figure 5.2.1: Steps of 3D model based Validation……..48

Figure 5.2.2: 3D Model created in 3D world Editor……49

Figure 5.2.3: View of 3D Model……..49

Figure 5.2.4: Dialog Box for Rotation of 3D Model……50

Figure 5.2.1.1: 3D Model…….....52

Figure 5.2.1.2: 3D Reconstruction of 3D Model…….....52

Figure 5.2.1.3: Best Gripping feature for grasping of 3D Model…….52

Figure 5.2.1.4: 3D Model……53

Figure 5.2.1.5: 3D Reconstruction of 3D Model……..53

Figure 5.2.1.6: Best Gripping feature for grasping of 3D Model…….53

Figure 5.2.1.7: 3D Model……54

Figure 5.2.1.8: 3D Reconstruction of 3D Model…….54

Figure 5.2.1.9: Best Gripping pattern for 3D model……54

Figure 6.2.1: (a) shows the input images taken from the camera (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object……56

Figure 6.2.2: (a) shows the input images taken from the camera at different angles, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object……..57

Figure 6.2.3: (a) shows the input images taken from the camera at different angles, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected
object……58
Figure 6.2.4: (a) shows the input images taken from the camera at different angles, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected
object……59
Figure 6.2.5: (a) shows the input images taken from the camera, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object……59

Figure 6.2.6: (a) shows the input images taken from the camera, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected……60

List of Tables

Table 2.2.5.3.1: Illustrating the success rate of 'Robotic Grasping of Novel Objects' 22

Table 3.7.5.1: Processing Time of SIFT and SURF for Feature Matching..... 40

Table 4.3.1.1.1: Length of Human Hand..... 46

Table 4.3.1.1.1: Breadth of Human Hand..... 47

Table 2.2.2.1: Result comparison of 3D model with software calculated values 55

Table 3.2.1: Comparison between Actual and Software calculated perimeters 60

Table 7.1.1: Comparison of our Research with other Recent Research..... 61

Chapter 1: Introduction

1.1. Introduction

Within the field of medicine, a prosthesis is defined as an artificial device that replaces a missing body part lost through trauma, disease, or congenital conditions. A prosthesis that replaces a part of the arm between the elbow and wrist is called a trans radial prosthesis, also referred to as a “BE” prosthesis for below-elbow. These devices can be functional or simply cosmetic depending on their intended use. An amputee who does a lot of manual labor and needs a device that is durable, dependable, and strong may choose to have a simpler prosthesis such as a hook. On the other hand, an individual who is willing to sacrifice functionality for a prosthesis that looks more natural may choose to get a cosmetic prosthesis, also known as a cosmesis, as one shown in figure 1.

Figure 3.1.1: Personalized Cosmetic Prosthetic Hand by Sophie de Oliveira Barata [17]

We developed an a prosthetic hand that is as natural and easy to use as a person’s organic extremity capable of determining the most suitable grasping pattern for griping an object and executing that grasping pattern with minimal human input without the need for invasive surgical procedure.

Functional Trans radial prostheses are available in two main types, body powered and externally powered. Body powered prosthetic limbs are controlled using cables connected to a harness or strap mounted elsewhere on the user’s body. When the user moves their body in certain ways they pull on 4 the cables to cause motion in the prosthesis. The simple nature of these devices makes them very light but also means they are typically incapable of executing

complex tasks. A common terminal device for body powered prostheses is a pincer like mechanism called a split-hook, illustrated in the diagram below, though many other's exist for more specific tasks like fishing or cooking

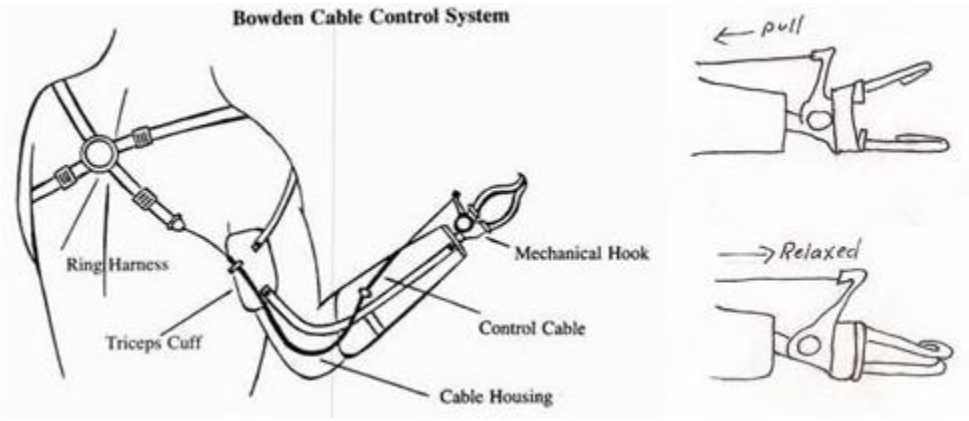


Figure 1.1.4: Typical Body Powered Trans radial Prosthesis [17]

Externally powered prostheses are devices that receive their power from sources other than the user's body. Usually electrically powered, modern devices are able to utilize multiple electric motors and other electrical components to achieve more complex grips and functionality than can be accomplished with a simple body powered device. The additional use of on board microcontrollers and sensors allow for these prostheses to be controlled in a variety of ways.

One common technique for controlling an externally powered prosthetic device is the switch control method. This method allows the user to move their prosthetic device by toggling switches or buttons. A user can toggle the switches using another part of their body such as their opposite shoulder, or with the remaining muscle in their residual limb. Since these devices are typically able to perform such a wide variety of grips, the user can often use different sequences of switch toggles to alternate between different grip modes.

Another, more advanced method of controlling an externally powered prosthetic device is through the use of electrodes. When placed on the surface of the skin, these sensors are capable of detecting the small electrical signals generated by muscle contractions in the user's residual limb. In most applications additional software and circuitry are used to make these analog devices behave like switches. The user is then able to control their device in a similar manor to the switch control method. Devices that utilize this technology are called myoelectric prostheses. Some examples of commercially available myoelectric prosthetic limbs are the Bebionic hand and the i-Limb. Each device is an advanced externally powered prosthetic limb and is currently considered top of the line in commercially available trans radial prostheses ("Myoelectric Prosthetics.", 2014).

1.2. Non-anthropomorphic prosthetic devices

One of the most common non-anthropomorphic terminal devices used on upper body prostheses today is the split-hook; a simple device primarily comprises of two hooks which are jointed together at the base by a hinge. This design enables the hooks to open and close in a pincer like fashion allowing for basic grip functionality. The curved shape of this prosthesis offers a fair bit of functionality as well, so long as there is a hole, handle, or divot for the hook to fit into. These devices are typically body powered though externally powered versions are available. Relative to other more versatile prosthetics, what the split-hook lacks in functionality it makes up for with durability and a low cost ("Prosthetic Devices.", 2014)

Figure 1.2.1: Common Split-Hook Terminal Prosthetic Devices [17]

Custom Non-anthropomorphic prostheses are also developed for more specific tasks such as cooking or fishing, or even for sports like basketball, climbing, or golf. It is not uncommon for an individual to own several functional prostheses intended for different tasks as well as a cosmesis for social events.

1.3. Anthropomorphic Prosthetic Devices

On the other side of the trans radial prosthetic spectrum are devices such as the i-LIMB and the bebionic hand. This new generation of externally powered robotic prostheses combines functionality with a natural, anthropomorphic appearance. The inclusion of an opposable thumb and four independently actuated fingers allows for not only more human like movement but also a wider range of grip patterns. Sensors in the device allow for system feedback resulting in better performance, and in some cases even user feedback through the use of lights, vibrating motors, or other interfaces. The prostheses are typically myoelectric. To operate the

device the user preforms combinations of muscle contraction that will initiate one of the preloaded grip patterns.

Figure 1.3.1: Anthropomorphic Prosthetic Hands i-LIMB Ultra Revolution & i-LIMB Cosmetic Cover [24]

Though these devices have a lot of advantages over earlier, simpler prosthetics, the additional weight caused by the onboard electronics can cause them to be uncomfortable to use for long periods of time. Another drawback of these devices is their high price. Peaking at about $100,000 after fitting and training, it is difficult for many potential users to afford one even with insurance

1.4. Problem Statement:

One of the unique creations in the human body is the hand which enables the human to grasp and manipulate objects. In both cases, people can sense and understand the environment. Unluckily, the people with upper limb disability cannot perform daily routine tasks properly which led them to the physical and psychological pressure.

Though the prosthesis industry has experienced a great revolution in upper body prostheses with the introduction of advanced myo electric grippers [1]. But the Users are still far from being able to perform their daily routine tasks with same level of ease as they could.

There are many products available in the market such as Bebionics The Hand and i-Limb ultra but these products have many complaints. Complaints of available products includes difficulty in performing daily tasks due to complex user interface and high market costs. Due to high market costs every one cannot purchase these products. The current statistics includes average of 18,496 upper-extremity amputations every year, compared to 113,702 of the lower extremity. Of those, only 1900 are above the wrist. Among upper-limb amputees, typically fewer than half wear prosthetic arms. An estimated number of 541,000 Americans were living

with some form of upper limb loss in 2005 and this number is projected to more than double with an aging and growing population by 2050. [1] When a person becomes a limb amputee, he or she is faced with staggering emotional and financial lifestyle changes. The amputee requires a prosthetic device(s) and services which become a life-long event.

These issues can be overcome by the implementation of a digital vision subsystem embedded in the prosthetic that will allow our device to determine the shape of the object the user is reaching for as shown in figure 1. The hand will then be able to take this information and automatically adjust to an appropriate grip. The end goal is that the user will only need to reach for an object and tell the device when to close. The process of selecting and executing a particular grasping pattern will be taken care of automatically, much like what is done naturally in our subconscious.

Figure 1.4.1: Pro E model Pro E model of Prosthetic hand with camera embedded in the palm of hand

Recognition and Reconstruction of object are two long standing issues in machine vision. The structure and movement (SAM) issue has achieved a level of development, with a few several commercial offerings [10] [22] in addition to comprehensive research literature [17] [8] [13]. Object Recognition is likewise very concentrated however; late advances in image features and probabilistic displaying have roused already unexplored territories, for example, object class recognition [7]. Invariant local features have developed as a precious device in handling the pervasive image correspondence issue. By utilizing descriptors that are invariant to translation, as well as to rotation [16] scale [9] and relative distorting [3] 12] [11], invariant features give a great deal stronger coordinating than past connection based techniques.

Recently algorithms have been developed that operate in an unsupervised manner on an image dataset. We operate in an unsupervised setting on an unordered image dataset and pose. Feature between all images are matched by using Speeded-Up Robust Features (SURF) and by using the RANSAC algorithm noisy matches are eliminated and find those matches that are consistent with the fundamental matrix.

1.5. Aims & Objective of the Thesis:

The aim of our research is, to develop a prosthetic hand that do not require complex circuitry to grasp an object. A hand that has a camera embedded in the palm of the Hand and capable of determining the most suitable grasping pattern for griping an object. A hand that requires minimum human input.

The objective include

- Single object Recognition using stereo vision
- Determination of Suitable Grasping Pattern in order to hold the object.

Our Research work has two parts; the first part of our research is 3D reconstruction of scene/ object from multiple input images taken from the camera. The second part of our research includes calculation of perimeter of detected object in world co-ordinates from 3D reconstruction and execution of best grasping pattern for the griping the detected object. *Chapter 06* demonstrates results of object 3D reconstruction and best grasping patterns for the detected object.

1.6. Thesis Overview

Chapter 2 – Advanced Myo Electric Commercial & Vision Based Research Prototype Hands

The chapter will give a brief overview of the background of the advanced myo electric grippers available in the market and Research prototype hands that have digital vision system embedded in the palm of the hand.

Chapter 3 – Methodology

The modules of our system i.e. image acquisition, basic image processing operation, noise reduction, camera calibration techniques , step of camera calibration, feature extraction methods, feature matching techniques, image matching, rejection of outliers, object recognition, and object reconstruction are discussed in detail in this chapter.

Chapter 4 – Object Identification and Execution of Best Gripping Features

In this chapter calculation of perimeters of the object, Human Hand Anthropometry and execution of best gripping features for the detected object is in detail..

Chapter 5 – 3D Model Based Validation

. In this chapter we will discuss how to create a 3D model, creating a dialog box to rotate 3D object at any angle and capture images at different angles, 3D object reconstruction and execution of best gripping feature for detected object in detail.

Chapter 6 - Experimental Results and Comparison with Actual Perimeters of the object

In this chapter experimental results of different objects are shown, comparison of experimental results along with the software calculated results are compared and execution of best gripping pattern are shown in the chapter

Chapter 7 – Conclusion & Future Work

The research will be concluded with theses chapters followed with the future recommendations

Chapter 2: Advanced Myo Electric Commercial & Vision Based Upper Limb Prosthetic Hands

2.1 Advanced Myo Electric Commercial Hands

2.1.1 Bebionics Myo Electric Prosthetic Hand

The Bebionic Myoelectric Prosthetic hand is one of the most developed prostheses that have intelligence in it. The hand itself has two different sizes. The medium size is based off of the size of an average adult male's hand while the large is just slightly bigger. This system has independent motors for each finger which allows for each finger to have its own unique configuration to the others. This allows for a large amount of varying grip types since each finger is able to move on its own. Two of these grips include power grip and a static hook grip. The time it takes to go from an open hand to power grip is 0.5 seconds and the maximum grip force that can be applied on an object is 140.1 Newton's. The power grip can be used to throw a ball, eat a piece of fruit or grip other spherical objects. The maximum static load that the hook grip is able to hold for the whole hand is 45 kg while an individual finger in hook grip can hold up to 25 kg. The hook grip can be used to carry something like a grocery bag or even grasp a door handle. Currently in order to change the thumb from an opposed position to a non-opposed position, the user must manually move it. Though to move from grip to grip, electrodes are used to allow the user to use their muscles to control the opening and closing of the fingers. The manner at which they apply these signals (short bursts, quick rising stimulus, slow falling stimulus) determines what grip is chosen. In addition to having separate finger control, there are three different wrist attachments available for the bebionic3 myoelectric prosthetic which can be seen in figure

Figure 2.1.1.1: Bebionic Wrist Options (small, medium and Large) [23]

The electric quick disconnect wrist (left) allows the wearer to quickly rotate and remove the hand to interchange with other terminal devices. This particular wrist only allows for a rotation using powered wrist rotators. The multi-flex wrist (right) allows the wearer movement in all directions giving it 3 degrees-of-freedom in all rotational directions and in addition to control or the roll, pitch and yaw of the wrist, the wrist can lock in 30 degree flexion, 30 degree extension or a neutral position. This allows the user to be able to perform daily tasks while not having to worry about the hand slipping or rotating in any direction. This more sophisticated version of the EQD wrist also still has the ability to be interchanged with other terminal devices. The final wrist type that the user has the ability to choose from is the short wrist (middle). This offers a reduced build height to accommodate for those amputees with amputations closer to the wrist and would not otherwise be able to support either other wrists naturally without leaving that particular arm long. This wrist choice only allows for rotation around the arm though not the other two degrees of freedom allowed from the multi-flex wrist.

This sophisticated prosthetic even allows users to decide whether they want a glove to give a natural look and feel made of silicone. This silicone is made using multiple layers similar to human skin.

2.1.2 i-Limb ultra

The i-Limb made by Touch Bionics is another example of an anthropomorphic myoelectric prosthetic device for hand amputees. Though it is very similar to the bebionic3, each finger is individually powered. The thumb must be manually rotated in order to achieve various grips. One key feature of the i-Limb is its proportional control. Like the bebionic3, this i-Limb is electrode controlled. The stronger the input signal that the electrode receives, the faster the fingers will move. This gives the user complete control over the speed at which the fingers both close and open their grip. In addition to this, if the user pulses the input, the group can increase the amount of force it has on the object though there is no real feedback that the user gets indicated how much force it is actually applying. The i-Limb is made of an aluminum chassis for increased durability and long life span. There are two available sizes, one corresponding to the average size of an adult male's hand, and one corresponding to the average size of an adult female's hand. One useful feature is that after long periods of inactivity, the hand will automatically move from whatever grip configure it may have been in to a natural position.

One key feature that truly sets the i-Limb apart from the bebionic3 is the ability to customize the configurations of the hand. Using the my "i-Limb" app that can be downloaded to any phone or computer for free, the user has the ability to not only customize the types of grips

most frequently used, but even customize specific gestures that are commonly used. This app can also control the performance output of the hand, the different grips in use at that moment, all the way down to the exact position of each individual finger. Two pictures from the mobile app of picking a grip as well as changing the finger position can be seen in figure

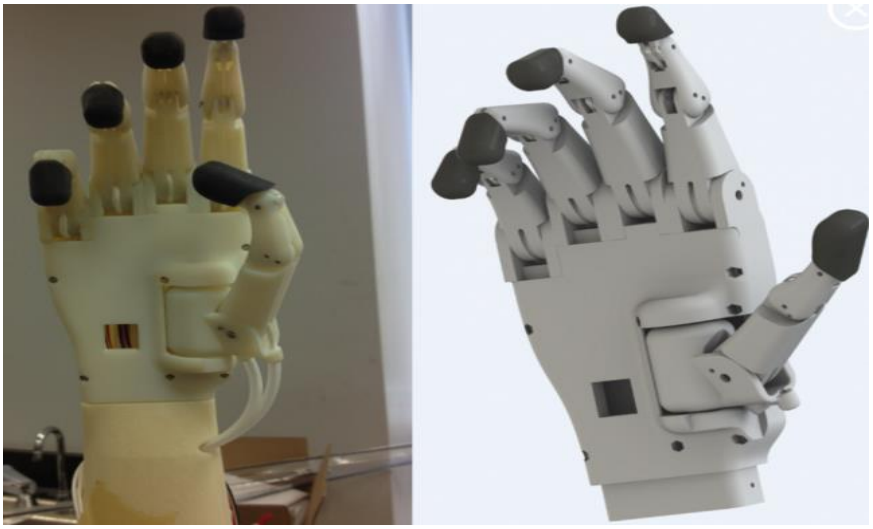
Figure 2.1.2.1: i-Limb Mobile App for interfacing [23]

In addition to these features, the i-Limb also has multiple available coverings. These coverings are to ensure that the user can use their device while handling dust as well as water and neither will affect the internal circuitry. Two popular non-human like coverings that the i-Limb have are the active skin and the active TS. These are made of semi-transparent, clear or black materials that are computer modelled to fit perfectly on the device. The difference between the two is that the TS version includes a conductive tip on the index finger making it compatible for touch screens on smartphones or tablets.

2.2 Vision Based Upper Limb Prosthetic Hands

2.2.1 The IRIS hand (2014)

Mounting a camera in the robotic/prosthetic hand is not a novel idea [17]. The IRIS hand that was developed in 2014 by WPI smart robotics lab (US) has a camera embedded in the palm of the hand as shown in figure.



(a) (b)

Figure 2.2.1.1: (a) shows the 3D print model (b) shows the CAD model of IRIS hand [17]

The object recognition of IRIS hand includes three steps Edge matching, Hough Transformation and SURF detection. Most essential edges from the input image are detected by edge matching object detection technique. Hough line transformation is performed on the image that generates a set of vectors that defines the edges and lines of the object [17].

SURF (Speeded Up Robust Features) Detection, works to detect similarities between two images by identifying unique features. Reference images are supplied, and the unique features are extracted. These features are then searched in the test image independent of the poses of the features in the image. The invariance of pose and scaling creates a very exceptional use of this technique, which allows the reference image to be identified in the test image along with its location in 2 dimensions, its potential distance from the camera, and its rotation and angle relative to the frame of the image.

Features are identified in the reference images by performing a series of Gaussian functions and detecting the differences following each subsequent Gaussian transformation. Areas that contain little to no change in appearance following each transformation are not considered as features in the reference image. Areas that demonstrate considerable contrast are blurred again in order to further determine the likelihood of features being present. These features, once identified in the original reference image, are searched for in the test image. Hough transformations are used (in a manner described similarly before) to determine potential clusters of features. Clusters of a specific feature with higher probabilities (determined by their location relative to other clusters of specific features) are granted a higher weight and likelihood of being present. The greatest

set of clusters, with a higher probability relative to each other cluster, is used to identify the object's presence and orientation in the image. Additionally, if the reference is image is scaled to a known size, the size of that reference image in the test image can be used to determine its distance from the camera.

From this method and the use of reference images, SURF detection can be used to identify known objects or images within other scenes. When compared to SIFT detection, SURF is much more robust and quick. This is largely due to the use of reference images to reduce computation time and increase the understanding of desired features. This method is useful for identifying complex flat images with great accuracy. However, it is not capable of understanding the 3D shape of objects given its 2D reference images. The object can only be identified given the specific angle and positioning in the reference image. For example, it can only understand a single perspective of a cube and has no understanding of what is on the other side of a cube. As such, without another reference image to fully describe the other side of the cube, it would not be able to properly identify it. In essence, this algorithm is heavily dependent on the presence of comprehensive reference images [17].

2.2.2 IRIS Hand Electrical Architecture

The IRIS hand electrical architecture involved a number of different pieces of hardware to achieve Communication between the different hardware, finger actuation, and object recognition.

2.2.2.1 Communication

At the highest level of the electrical system, the team used an Arduino Pro Micro board to communicate between the various components, and act as the brain of the system.[17]

This board was responsible for sending signals that would actuate the motors, read in motor positions for positional control and force control, and interpret data from the object detection system. The Pro Micro communicated directly with a 16 channel Analog Multiplexer (MUX) which was capable of reading and writing to 16 analog inputs or outputs using only 5 pins from the Arduino.

Similar to the MUX, a 16 channel PWM driver was used to send out a total of 16 Pulse-Width Modulated (PWM) signals to the motor drivers which can then use those signals to drive the motors. The PWM driver allowed for these 16 signals to be sent simultaneously using only two pins on the Pro Micro that setup an I^2C communication using features of the Arduino IDE.

2.2.2.2 Finger Actuation

Utilizing both the MUX and the PWM driver, the finger actuation was achieved. As mentioned the PWM driver was able to send out a total of 16 PWM signals. The dual motor driver was connected to four of the PWM Driver pins. In order to drive a single motor two of the pins needed to be connected to the motor inputs for one motor on the dual motor driver allowing for speed and directional control of that motor.

The motors driven by the dual motor driver were the Pololu Micro-gear motors with a 1000:1 gear-ratio, allowing for high torque output from compact motors. These motors drove the series-elastic-actuated system by rotating the pulleys to actuate the four-bar linkages driving the fingers.

Though communication with the motor driver through the PWM driver allowed for both the speed and directional control of an individual motor, the scope of the project required control of the position of the motor. To achieve this the team used rotary potentiometers which would vary their resistance based off of the angle that their shaft was rotated.

These potentiometers were used in the series elastic actuation system to determine the position of the motor shaft and the spool connected to the finger allowing for calculations of the force on the fingers to be made based off of the offset between the two potentiometers.

2.2.3 Limitations of IRIS Hand

The limitations for of IRIS hand are that;

- Edge matching technique is requires a comprehensive set of reference images that compare with the image from a camera. So, IRIS hand requires a lot of memory to save these reference images.
- Best Gripping feature is decided on the base of number of detected lines. If detected lines are more than 10 lines then the object is unidentified. If detected lines are less than 10 lines and more than 75% of lines are parallel or there are 3 or more parallel line groups then the object is cube.
- If there are 2 lines parallel group and have the same length then object is cube, if not, then the object is cylinder.

- If less than 75% of lines are parallel with another line than perform Hough circle transformation, if image contains more than one circle then the object is a sphere. If not, the object is unidentified.
- SURF detection is used to perform for determining the rotation and translation of the object from the camera.
- IRIS hand is not capable of identifying complex shape identifying cube, cylindrical and spherical shapes. Another disadvantage of this system that it only executes tripod grip for each detected cube and power grip for each sphere.
- Another disadvantage is that it only detect green color object.

Different grasping features of IRIS hand are shown in figure given below;

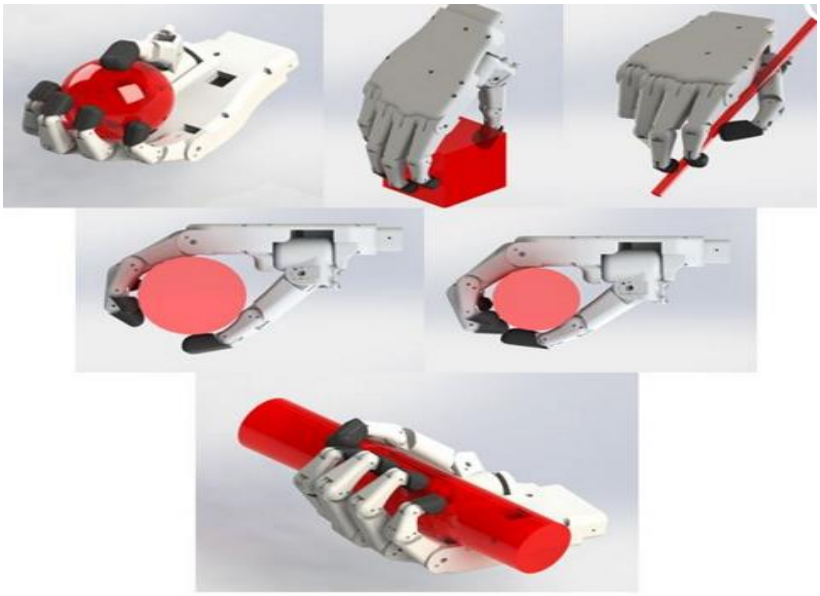


Figure 2.2.3.1: Different Grasps of IRIS Hand [17]

2.2.4 Vision-based Intelligent Prosthetic Robotic Arm (2015)

Prosthetic devices have been developed for many decades to assist and compensate people who have lost their limbs to accidents or pre-existing disabilities. Many prostheses have tackled problems on how to achieve similar if not parallel capabilities to those of human limbs. With increasing complexity and functionality of modern prostheses, control architectures have evolved to make operation of the device more intuitive and convenient for amputees in their daily routines. Examples of these control architectures include passive motion of the body and electrical signals sent to muscles. [23]

The previous iteration of this project, the IRIS Hand, involved the development of a smart robotic prosthetic device that utilized object recognition with a camera embedded within the device to identify objects and autonomously determine grasps prior to manipulating those

objects. As a continuation of this technology, this project involves the realization of a prototype trans-radial prosthesis. This includes the redesign of the actuation system in the finger and the mechanical design for wrist's motion.

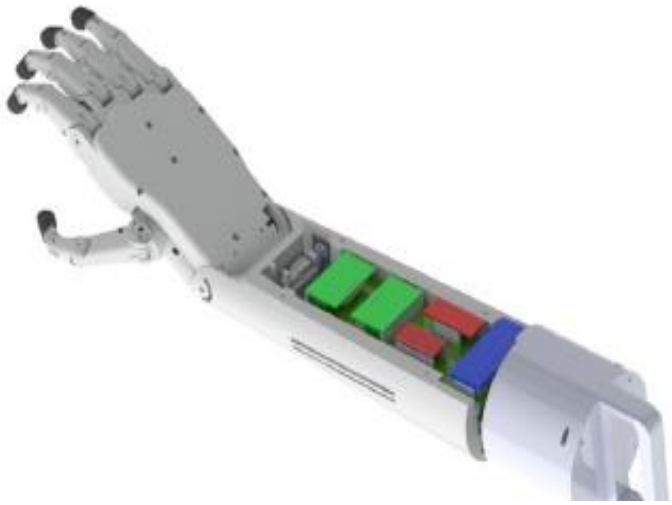


Figure 2.2.4.1: Vision Based Intelligent Prosthetic Robotic Arm [23]

Figure 2.2.4.2: Hand Case Design [23]

2.2.4.1 System Overview

Based on our previous research, it was decided the functionality of VIPeR Arm. Using the IRIS hand as a base design, a trans-radial prosthesis with six degrees-of-freedom in the palm and two degrees-of-freedom in the wrist was developed. The actuation of each finger was driven by a bevel gear system. One degree-of-freedom in the wrist used a belt and pulley while the other degree-of-freedom was actuated by a directly driven motor. We were able to measure the force exerted by each finger using the current spike from the motors. To achieve autonomous grip selection, a camera system located in the palm was used to detect AR markers corresponding to different objects.[23]

2.2.4.2 Analysis of IRIS Hand System

The Iris Hand electrical system components all were logical choices for the desired task. One issue the team had with their motor drivers was that they would burn out when the motors stalled. Upon further investigation we learned that the stall current of the 1000:1 pololu micro motors was 1.6 amps which is greater than the free run current of their motor driver which was 1.2 amps with a peak of 1.5 amps. We theorized this discrepancy was the main reason for their motor drivers burning out. We also saw that the old camera used in the arm was no longer functional. Majority of the components in the electrical system were all still fully functional, so with the change of the motor driver, addition of some form of current sensing, replacement of the old camera, and the addition of a higher torque motor for the wrist roll rotation, we were able to create the electrical architecture for the VIPeR Arm.[23]

2.2.4.3 Electrical Architecture

One of the major focuses of the project was to develop an electrical architecture that allowed for force sensing based on the electrical current in the motors when torque was applied. In order to determine how to best incorporate this in our system, we started by analyzing the electrical components used in the project last year to see which components could be recycled for our system. [23]

2.2.4.4 VIPeR Arm electrical Architecture

Due to all of the electronics in our, system, the circuit requiring all of them to communicate involves multiple connections. The diagram below shows the number of connections involved in the circuit excluding power and ground which are connected to an external power supply that runs the system.

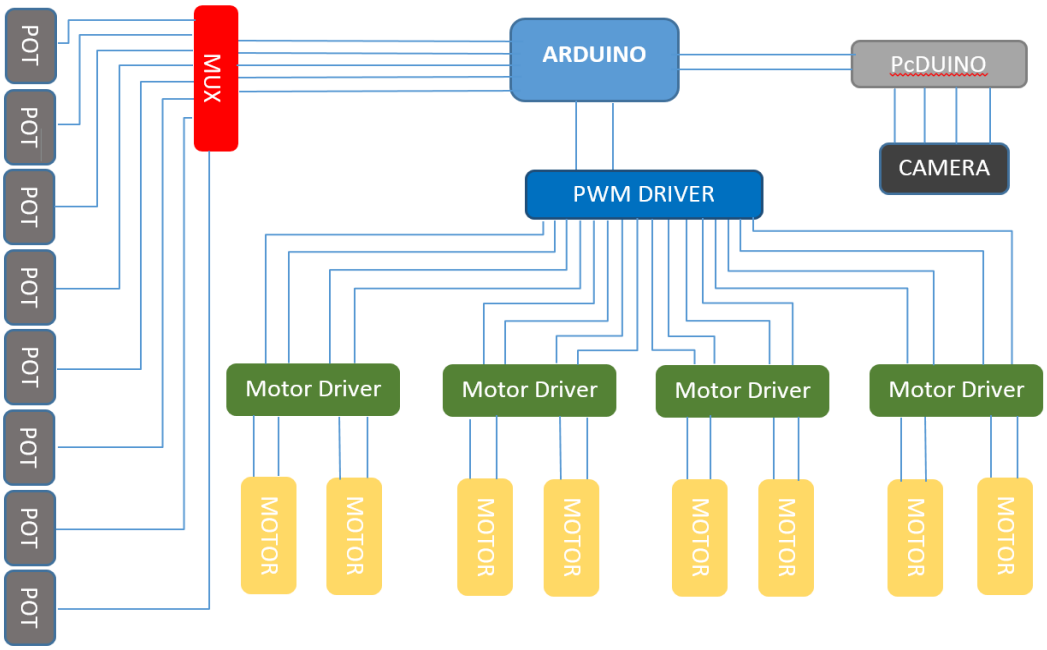


Figure 2.2.4.4.1: Electrical Architecture of VIPeR Arm [23]

Each component integrates into our electrical system to create our control loop that actuates the fingers based off of the AR tag recognized the camera. The system diagram below shows the full functionality of the system driven by the Arduino main processor.

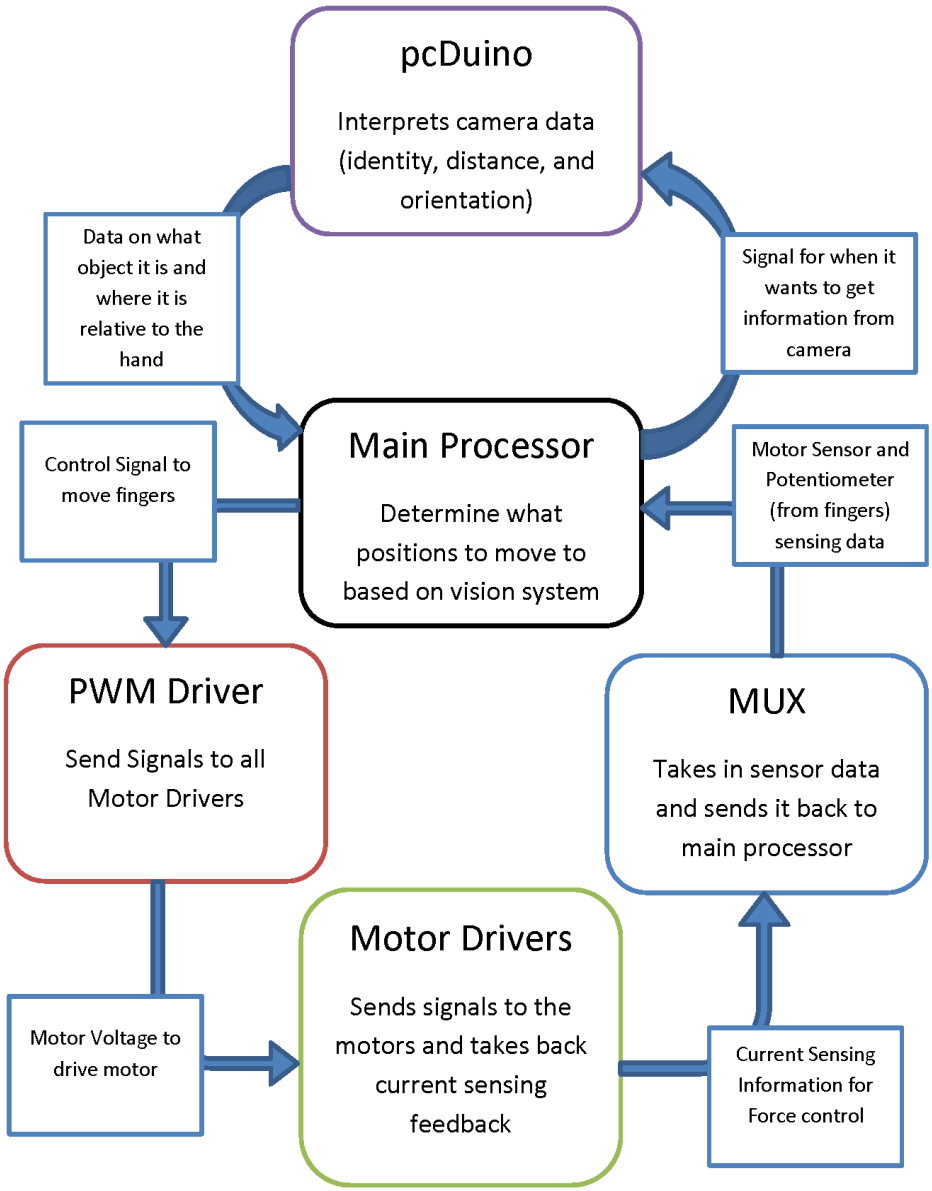


Figure 2.2.4.4.2: Flow Diagram of VIPeR Arm [23]

2.2.4.5 Object Recognition Components

Due to the large amount of time it takes to process data from a camera, the team last year decided to incorporate the use of a second device to process the camera data, and then send it to the main processor. The PcDuino was an ideal choice because, it was a mini computer able to Run Linux allowing for the PcDuino to download the necessary software to communicate with the computer and run edge detection algorithms for object recognition. The PcDuino communicated with the Pro Micro using serial communication pins on both boards.

The last component in the object recognition system was the camera which sent data over to the PcDuino to be interpreted. The camera was stripped down to allow for it to fit within the palm.

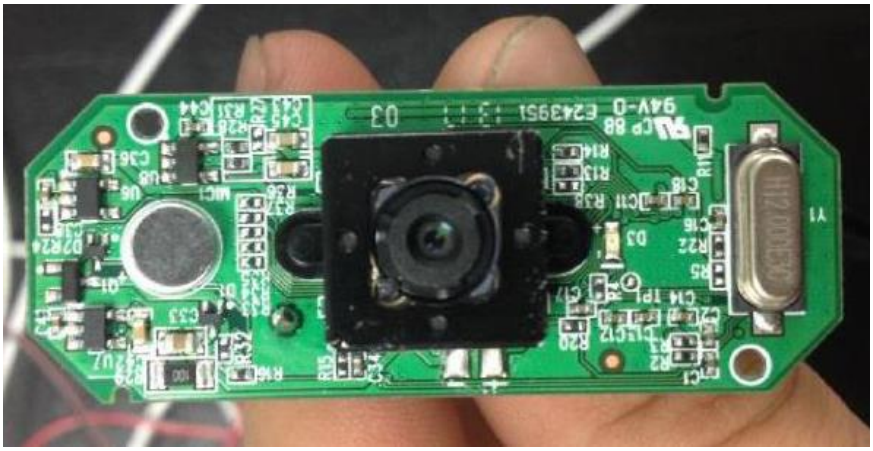


Figure 2.2.4.5.1: Digital Vision Module of VIPeR Arm [23]

2.2.4.6 Limitations of VIPeR Arm

- This system requires too much memory for storing reference images
- We cannot predict the 3D model of the object.
- We can execute the appropriate grip for grasping the object
- All objects are required to use an AR tag, meaning that heavy environmental modifications will have to be made for prosthetic users to utilize the system as efficiently as a user with a healthy hand would.
- It has a camera in the middle of the hand so this system cannot work when grip is closed
- This system cannot detect the objects of all colors.

2.2.5 Robotic Grasping of Novel Objects

Modern-day robots can be carefully hand-programmed or “scripted” to carry out many complex manipulation tasks, ranging from using tools to assemble complex machinery, to balancing a spinning top on the edge of a sword [Shin-ichi and Satoshi, 2000]. However, autonomously grasping a previously unknown object still remains a challenging problem [24]

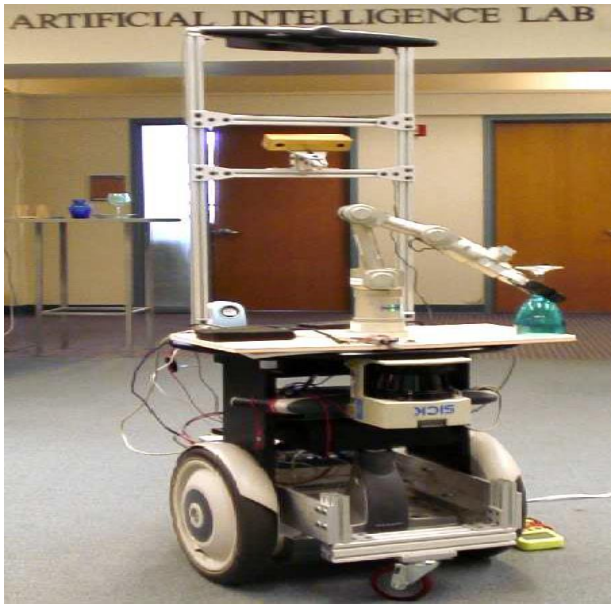


Figure 2.2.5.1: Robotic Grasping of Novel Objects [24]

2.2.5.1 Learning the Grasping Point

We consider the general case of grasping objects—even ones not seen before—in 3-d cluttered environments such as in a home or office. To address this task, we will use an image of the object to identify a location at which to grasp it. Because even very different objects can have similar subparts, there are certain visual features that indicate good grasps, and that remain consistent across many different objects.

For example, jugs, cups, and coffee mugs all have handles; and pens, white-board markers, toothbrushes, screwdrivers, etc. are all long objects that can be grasped roughly at their mid-point.We propose a learning approach that uses visual features to predict good grasping points across a large range of objects.

In our approach, we will first predict the 2-d location of the grasp in each image; more formally, we will try to identify the projection of a good grasping point onto the image plane. Then, given two (or more) images of an object taken from different camera positions, we will predict the 3-d position of a grasping point. If each of these points can be perfectly identified in each image, then we can easily "triangulate" from these images to obtain the 3-d grasping point. (See Figure.) In practice it is difficult to identify the projection of a grasping point into the image plane (and, if there are multiple grasping points, then the correspondence problem—i.e., deciding which grasping point in one image corresponds to which point in another image—must also be solved). This problem is further exacerbated by imperfect calibration between the

camera and the robot arm, and by uncertainty in the camera position if the camera was mounted on the arm itself.

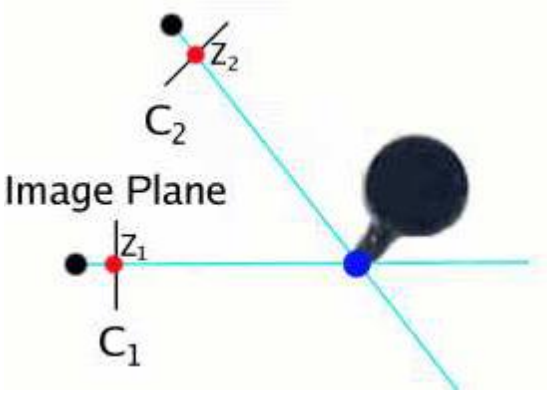


Figure 2.2.5.1.1: Intersection of Rays from images at Grasping point [24]

To address all of these issues, we develop a probabilistic model over possible grasping points, and apply it to infer a good position at which to grasp an object

2.2.5.2 Grasping Point

For most objects, there is typically a small region that a human (using a two-fingered pinch grasp) would choose to grasp it; with some abuse of terminology, we will informally refer to this region as the "grasping point," and our training set will contain labeled examples of this region. Examples of grasping points include the center region of the neck for a martini glass, the center region of the handle for a coffee mug, etc. (See Figure.)

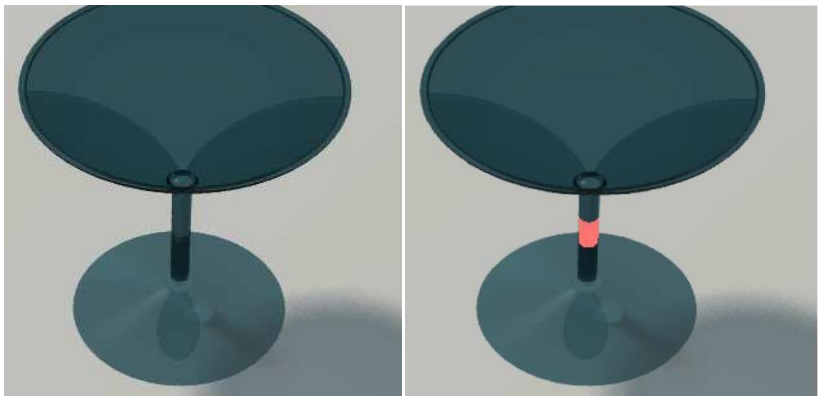


Figure 2.2.5.2.1: The red points in each image show the locations most likely to be a grasping point [24]

2.2.5.3 Results of Novel Objects

Objects	Error	Success Rate
Stapler	1.9	90%
Duct Tape	1.8	100%
Keys	1.0	100%
Marker	1.1	100%
Tooth Brush	1.1	100%
Jug	1.7	75%
Coiled Wire	1.4	100%

Table 2.2.5.3.1: Illustrating the success rate of 'Robotic Grasping of Novel Objects' [24]

2.2.5.4 Limitations

- It can create the 3D model of the detected object but the problem is that it always tries to find out the region that has smallest area and execute pinch grip to grasp the object.
- Another disadvantage is that it cannot execute other grasping features as humans do.
- It can grasp heavy objects

  So to overcome these issues discuss in literature review we proposed a research idea to develop a prosthetic hand that has a digital vision system embedded in the palm of the hand at thenar muscle, that requires minimum human input, capable of determining the most

suitable gripping pattern for gripping the object, that can execute different grasping features as humans do, that has low processing time.

2.2.6 PR2 Robot

The PR2, a research robot built and developed by Willow Garage, commonly uses a Tabletop Object Recognition package that employs model database analysis. The PR2 is equipped with a Microsoft Kinect sensor, which houses an RGB camera, a depth sensor, and an infrared projector. Using this device, the PR2 can acquire a 3D image of the world before it. The Tabletop Object Recognition package is used primarily for identifying objects on a flat tabletop-like surface. Objects that protrude from this surface are isolated from the table surface. These objects are one-by-one compared to objects in the 3D model database using eigenvectors to determine position and orientation of the object. Objects with a high similarity to known 3D models are compared until a certain similarity threshold is passed. Once this happens, the object is identified as the same 3D model it with which it was matched. [25]

However, this method does not work well with objects of an irregular shape or objects that do not match the known database.

This method can be a rather effective solution to the object detection. Its effectiveness is limited by the number and variety of object models stored in the database, the thresholds used to compare objects meshes, the sensing capabilities of the 3D camera hardware and software, and the processing power of the device that drives it. Also, it can only operate in specific environments that allow for better segmentation of potentially identifiable objects. Additionally, 3D cameras are often computationally expensive to operate at a useful frame rate, requiring a computer with more significant processing power. However, within it’s ideal environment and given the proper harder, with objects that do exist in its database, this system can reliably identify the object as well as determine its position and orientation relative to the camera

Figure 2.2.6.1: PR2 Robotic Hand [25]

Chapter 3: Methodology

3.1 Overview

In this section, algorithm and methodology of our system is discussed. The modules of our system i.e. image acquisition, basic image processing operation, noise reduction, camera calibration techniques , step of camera calibration, feature extraction methods, feature matching techniques, image matching, rejection of outliers, object recognition, and object reconstruction are discussed in detail in this chapter.

3.2 Flow Diagram of Our System

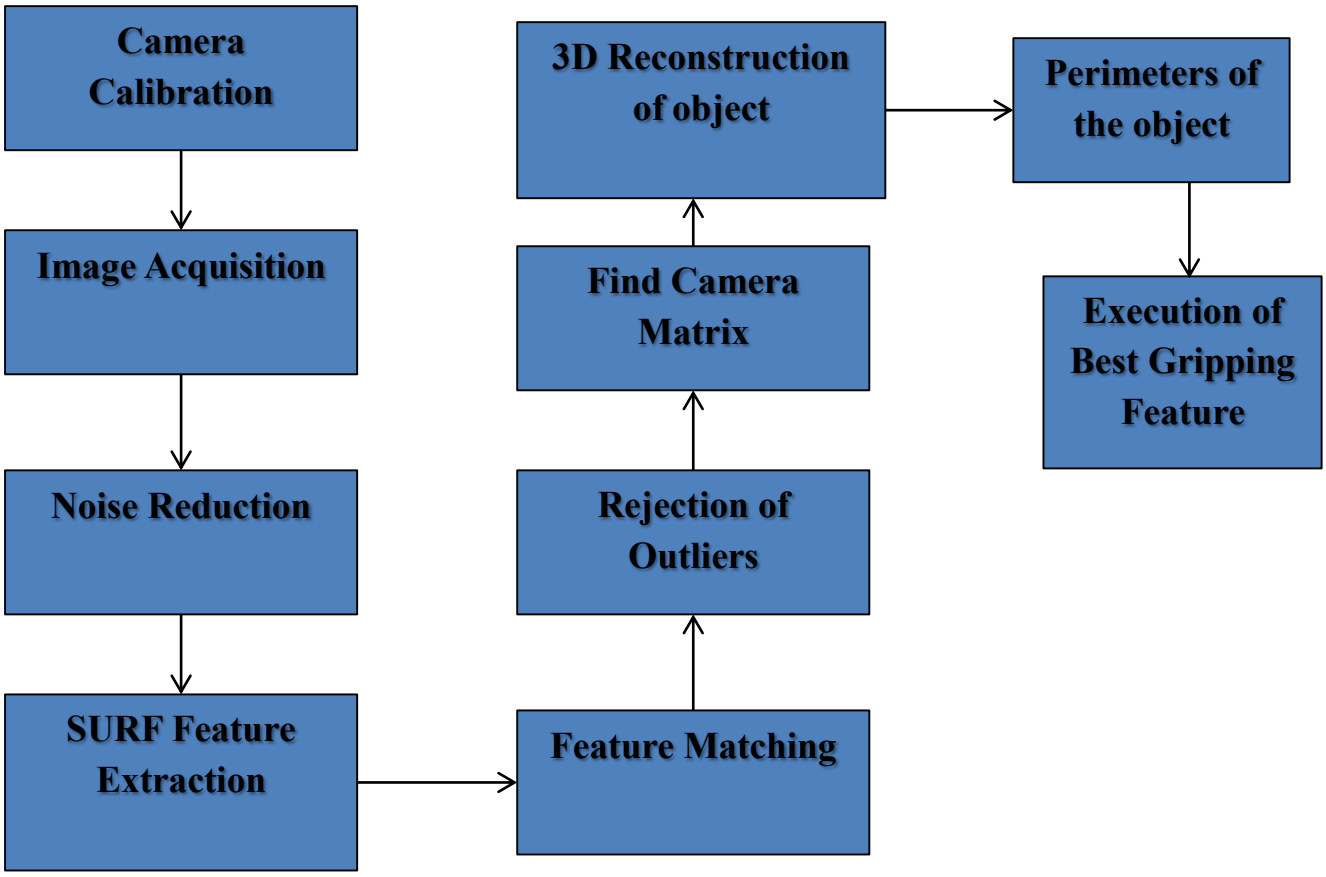


Figure 3.1.1: Flow Diagram of our Research.

3.3 Modules

There are two modules of our system shown in figure below

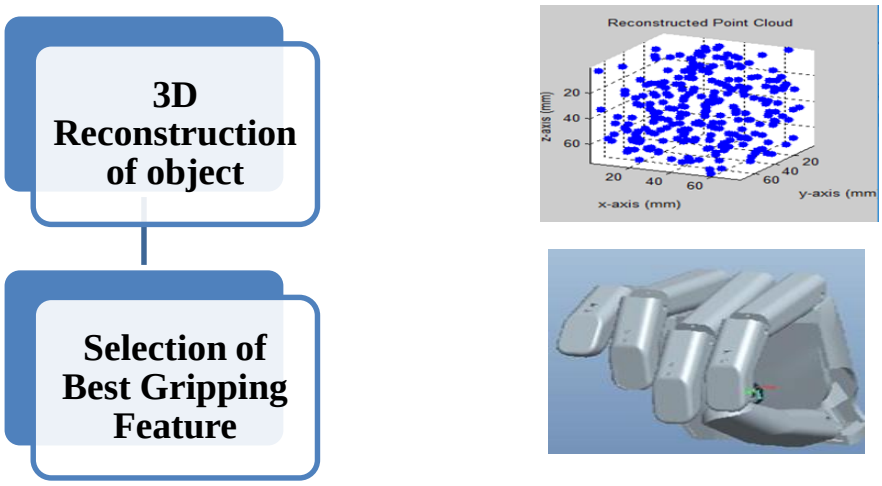


Figure 3.2.1: Modules of our Research

3.4 Image Acquisition and Noise Reduction

As Digital Image device (Camera) is embedded in the palm of a Prosthetic Hand that acquires multiple views of an object by using image acquisition toolbox built in MATLAB Software.

All input image are RGB images so for basic image processing first we convert all the RGB images into Gray scale images and then gray scale image to binary image. After converting all images into binary images we find number of connected regions in all the images. By applying thresh holding function we discard all those regions that have less than 1000 pixels as shown in figure given below.

Figure 3.3.1: RGB input image taken from image Acquisition Tool Box

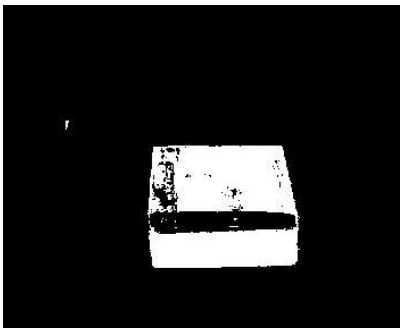


Figure 3.3.2: Conversion of RGB image to Binary Image

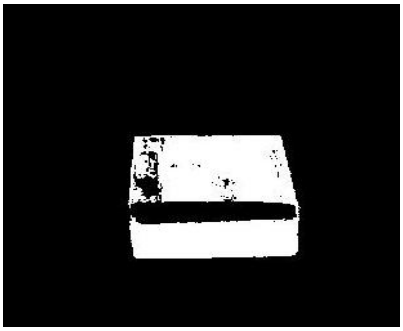


Figure 3.3.3: Binary Image After Discarding unwanted

3.5 The model of imaging geometry

Consider an orthogonal inertial reference frame {0;X;Y;Z}, called the world frame. Co-ordinates of a point p in the image the world frame can b written as [21]

$$Xw = [x_w, y_w, z_w] \in R^3$$

Where Xw is the world frame

In order to write the coordinates of the same point with respect to camera frame c the coordinates in the world frame and those in the camera frame are related by:

$$Xw \;\; = \;\; G_{\;c}^{\;w} \left[\; Xc \; \right]$$

Where G_c^w is called the transformation matrix and it can be described as

$$G_c^w = [R_c^w, T_c^w]$$

Where R_c^w is called the rotation matrix and T_c^w is called translation intrinsic parameters

In order to find intrinsic parameters of camera we specified x and y in terms of metric units and xs, ys coordinates of scaled version that correspond to the index of a particular pixel, then the G_x^{xs} can be described by a scaling matrix. Where G_x^{xs} called transformation matrix from x is coordinates to xs coordinates

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} sx & 0 \\ 0 & sy \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Now, projection model is combined with scaling matrix and translation matrix we get a transformation matrix. That transformation matrix transforms homogenous coordinates of 3D point with respect to camera frame. Homogeneous coordinates of its image expressed in terms of pixels can be written as:

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} fsx & fs\theta & 0x \\ 0 & fsy & 0y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$$

Where,

f is called the focal length of camera, fsx is called size of the focal length in horizontal pixels, fsy is called size of the focal length in vertical pixels, [xim; yim]T are called the image coordinates [21]. The intrinsic Matrix of camera is

$$\boldsymbol{f} = \begin{bmatrix} \mathbf{684.4} & \mathbf{0} & \mathbf{317} \\ \mathbf{0} & \mathbf{685.8} & \mathbf{246} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

3.6 Camera Calibration

From the above discussion it shows that the geometry of the camera that depends upon number of factors such as focal length, position of principle point and intrinsic parameters of the camera. In case the internal parameters of the camera are available, given two views of the

scene and number of corresponding points in both views, the Euclidean structure of the scene as well as displacements between the views can be recovered. This chapter will again strive for the recovery of the Euclidean structure of the world and camera pose in the absence of knowledge of internal parameters of the camera.[21]

The natural starting point is the development of techniques for estimating the internal parameters of the camera and reducing the problem to the one described in the previous chapter. Hence we first review the most common technique for camera calibration, by using a known object (a so-called calibration rig"). Since this technique requires a calibration object it is suitable only for laboratory environments, where both the camera taking the images as well as the calibration object are available. The second class of techniques for estimating camera parameters does not have this requirement, and involves estimating intrinsic parameters as well as scene structure and camera pose directly from image measurements; hence, it is called camera self-calibration. As we will see, given the richness and the difficulty of the problem, there are several avenues to pursue. We first focus on the so-called intrinsic approach to the problem of self-calibration which leads to analysis, algorithms and sufficient and necessary conditions for the recovery of the all the unknowns based on intrinsic constraints only. For the cases when the conditions are not satisfied we will characterize the associated ambiguities and suggest practical ways to resolve them. If we could measure metric coordinates of points on the image plane (as opposed to pixel coordinates).

$$\lambda x = PgX$$

Where P = [I,0] and g is the pose of the camera in the world reference frame. In practice, however, calibration parameters are not known ahead of time and, therefore, a more realistic model of the geometry of the image formation process takes the form

$$\lambda x' = APgX$$

where A is the camera calibration matrix that collects the unknown parameters.

$$A = \begin{bmatrix} fsx & fs\theta & ox \\ 0 & fsy & oy \\ 0 & 0 & 1 \end{bmatrix}$$

When we don't have any knowledge of internal parameters of camera, Camera calibration is used for the recovery of world structure and position of the camera.

The most common technique is calibration Rig so we used calibration Rig technique in our research.

3.6.1 Camera Calibration with Rig

For this type of technique camera calibration is done by using checker board patterns fixed on one or more planes. For calibration we used camera calibration application built in MATLAB R2014b software. By using this camera calibration application we convert the pixel information into the world structure and also find out the camera position and orientation as shown in figure

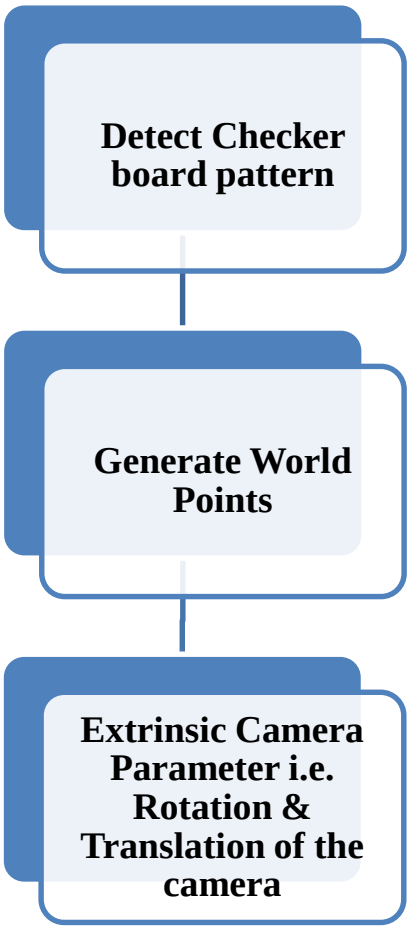


Figure 3.6.1.1: Steps of camera calibration with Rig

3.6.2 Results of Camera Calibration

For camera calibration the checker board we used is shown in figure below and size of one side of each checker board square is =23mm

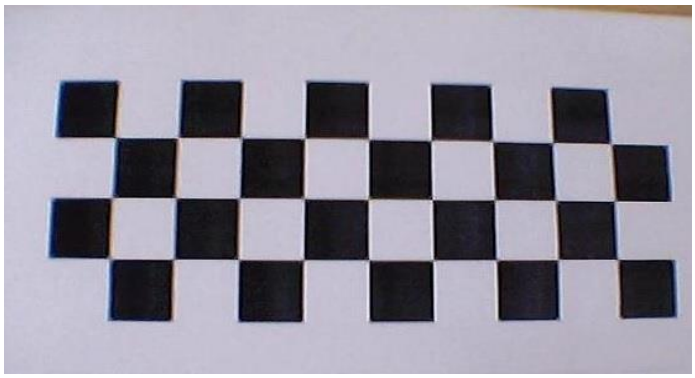
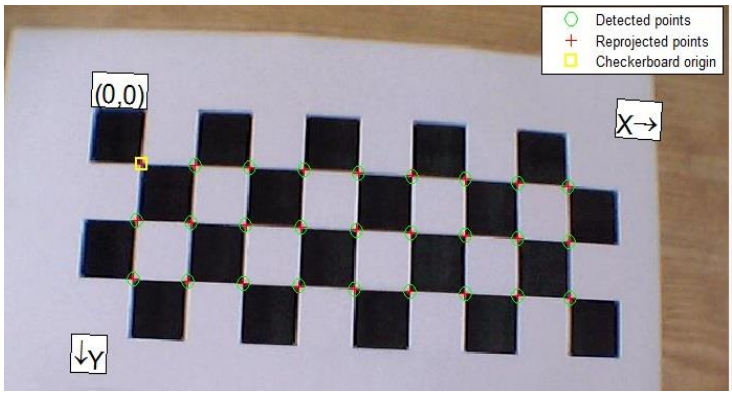


Figure 3.6.2.1: Steps 01: Detection of Checker board square



.

Figure 3.6.2.2: Steps 02: Generation of World points

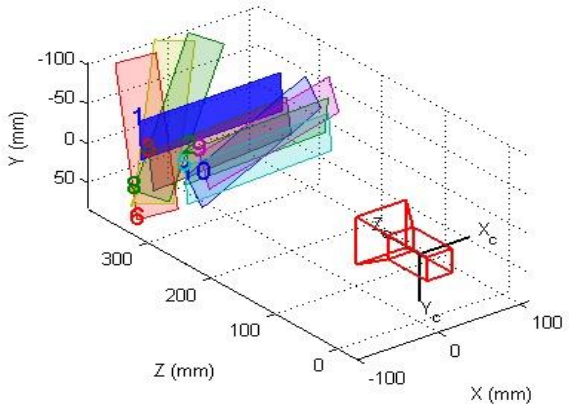


Figure 3.6.2.3: Step 03: Extrinsic Perimeter Visualization

3.7 Object Recognition Methods

Object recognition is an application of computer vision. Video recorded from a camera is parsed through a recognition algorithm to generate useful data, generally lines, geometry or points associated with objects within the camera's view. Recognition can be divided into two major categories: specific and generic. The generic method utilizes categorical traits of a type of item, and compares the image read against these categories to determine the type of object being viewed. The specific category, however, attempts to categorize an object as a specific item, such as a particular person's face or a particular structure; the bounds of the recognition are more strictly defined to determine the likelihood of the object being identified (Grauman, 2011).

For our robotic prosthesis, object recognition will be used to identify objects and determine appropriate grasps to lift or grasp the items that the arm is expected to interact with. To do this, further research would need to be done into the various recognition algorithms to determine the most appropriate direction. [23]

There are two major methods of object recognition:

- Appearance Based Object Recognition Method
- Feature Based Object Recognition Method.

3.7.1 Appearance-based Recognition

Appearance-based object recognition is the utilization of example images, which are used to compare against the live data set pulled from the camera input to determine likelihood of identification.

3.7.1.1 Direct Correlation Method

The direct correlation, or template matching, method is a comparison algorithm that utilizes stored images and compares the captured image data to that particular image. This data in some instances can be converted to a vector and directly compared to a vector generated from the templates (Heseltine, 2013). Through this, image similarity can be obtained and if the threshold of the algorithm for the difference in the two image vectors is not surpassed, the object is identified.

Some of the more primary concerns of the algorithm are variances due to lighting, orientation of the object, or distortion due to viewpoint or illumination. This can be accounted for through the use of multiple template images to correlate to different instances, or through applying a Gaussian blur to the template image to allow for a more coarse comparison (Berg, 2005).

3.7.1.2 Edge Detection

Edge detection is used to determine edges within a captured image of an object. These edges can be used in software to simplify a digital image grabbed from a camera or file system down to a collection of pixels representing edges, to put through additional filters. One of the more commonly used edge detection methods is the Canny Edge detection algorithm (Karla, 2009). The focus of the Canny edge detection is to maximize the detection of real edge points and lowering the probability of detecting non-edge points (false positives), and only detecting one edge per real edge in the image (Karla, 2009).

The approach to a Canny edge detection algorithm is as follows:

1. Initial conversion of the image to grayscale. Histogram-stretching to utilize full gray-scale range (this step may not be completely necessary depending on the tuning of the constants in your algorithm).
2. Application of a Gaussian filter, or blur, in order to reduce potential noise from the system.
3. Calculation of the gradients of the image. This is done through the use of the Sobel-operator, which performs a 2-D spatial gradient measurement (Karla, 2009). This can be used to determine direction of changes in the gradient.
4. Non-maximum suppression. All local maxima within the gradient image are preserved, while removing other values. The thick approximate lines generated from the gradient calculation are thinned to finer edge approximations.
5. Double thresh holding. Two thresholds are used to mark strong, weak, and negligible edges. Only strong and weak edges are preserved, while negligible edges are deleted.
6. Edge tracking by hysteresis. Only weak edges connected to strong edges are preserved, while isolated weak edges are removed from the model.

From this, the image has been refined and isolated to only a collection of edges, which can be used by the device for further analysis. Canny edge detection is often used in conjunction with other imaging software to simplify input images for comparison to templates with reductions

in effects from illumination or noise. This edge detection model is useful due to the use of two thresholds to determine important lines, which allows for more refined parsing of the image.

3.7.2 Feature-based Recognition

Feature based object recognition is the extraction of features from the contours and data of an image. These features are comprised of lines, arcs and lobes, which can be refined further and combined to make larger features (Howarth, 2009). From this reduction an input image can be reduced to a series of shapes or features, which can be used to determine the geometry of an object. These can generally be used in conjunction with appearance based recognition methods and a database of images and templates to determine the most likely object held within the image.

3.7.2.1 SIFT Detection

Scale-Invariant Feature Transformation, or SIFT, is a descriptor that is calculates points of interest within an image, utilizing directions of local gradients. At each of these points of interest, image descriptors are calculated, which are normalized to be scale-invariant and rotation-invariant. These points of interest are then used to compare and match to points of interest generated within other images.

From here, local image descriptors between two images can be matched to perform object recognition using existing templates; these matches can be visualized as a line between each match, which takes into account changes in scale or orientation of the item within the image. An example of two images being compared. There are a few methods for determining point matches within two images, such as the application of a Best-Bin-First (BFF) algorithm to scale for larger numbers of image descriptors (Lindeberg, 2012).

3.7.3 Fiducial Markers

Fiducial are small markers that are used in both human sight applications and computer vision systems to obtain data of an object or environment (Fiala, 2004). For computer vision, a code base utilizing a computer vision library such as OpenCV can read in the image and parse for the desired information. Fiducial markers are used in several industries, such as for industrial use, commercial use and other position tracking systems. An example of some types of marker patterns can be seen in figure

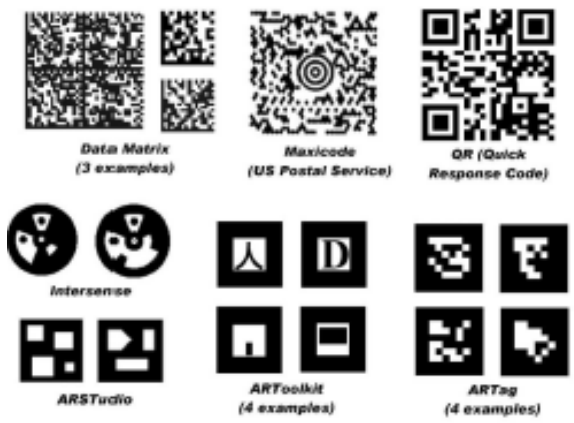


Figure 3.7.3.1: Several Types of Planar Pattern Marker Systems (Fiala, 2004)

Using custom fiducial patterns, it is possible to store information that can be queued or utilized by a fiducial. One such application of this feature is augmented reality. Through processing of the image for the fiducial, orientation and position of the fiducial with respect to the camera can be determined (Fiala, 2004).

The ARTag system is an example of a fiducial marker system utilized for generation of augmented reality (Fiala, 2004). This system utilizes a 36-bit code through use of a 6-by-6 grid within a bounding area. The values of the grid can be parsed and read in as a 32-bit value. The ARTag is first parsed using a form of object recognition, where after a quad is generated denoting the position and orientation of the identified fiducial (Fiala, 2004). From this, the value of the tag is stored at that location in the camera. A main application of this software is 3D model localization for augmented reality software. However, this could be applied to a prosthesis or robotic arm to determine the position and orientation of an object stored in the system.

ARToolkit is another fiducial-based marker system, utilizing a more varied set of input markers, such as symbols, letters and more abstract fiducial patterns.

The advantage of this system is that many assumptions about the object that we are visualizing can be made based off of the orientation and position of the fiducial tag with respect to the camera. This means that computer generated graphics could be rendered in real time overlayed in a recorded environment. This could also be used to approximate the position of an object given that the robotic system reading the tag can pull the associated object from a database. One potential issue with the markers in this use case is that, while accurate in finding the tags, there are still many assumptions being made on the system; it is assuming that the object always

has a consistent geometry and does not deform. Though a safe assumption depending on the object being identified, this would limit the mechanism's capabilities of what types of object it could handle. If used in conjunction with more appearance-based, or feature-based, recognition systems, non-standard objects could be accounted for. However, that would require testing of the system to understand the feasibility.

3.7.4 SURF Feature Extraction

After image acquisition and performing basic image process the next task is to extract SURF features from the input images.

Speeded Up Robust Features (SURF) is a local feature detector and descriptor which is partially motivated from the Scale-Invariant Transform (SIFT) that can be used for recognition of the object, classification, finding correspondence points between multiple images of same scene or objects and object reconstruction [17].

3.7.4.1 Steps of SURF Feature Extraction and matching

Following are the steps for extracting SURF (speeded Up Robust Features) and feature matching shown in figure below.

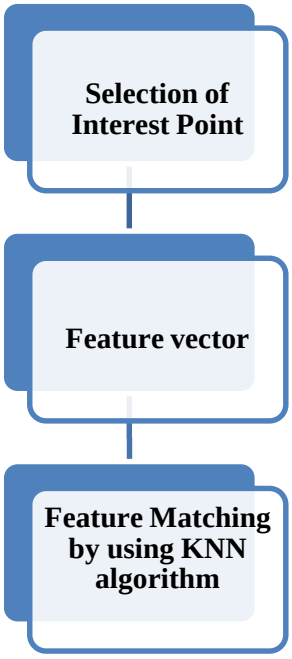


Figure 3.7.4.1.1: Steps for SURF Feature Extraction and Matching

There are three steps for finding image correspondence points. The first step is selection of at `interest points' at different locations in the image, such as corners, blobs, and T-junctions. The repeatability of interest point detector is valuable because it tells us the reliability of a detector for finding the same physical interest points under different viewing conditions.

The second step is to represent a feature vector of every nearby interest points. This descriptor has to be distinctive and at the same time robust to noise, detection displacements and geometric and photometric deformations. For this we use very basic Hessian based approximation. Finally step is the matching between the descriptor vectors of different images.

3.7.4.2 Results of SURF Feature Extraction and matching

Figure 3.7.4.1.2: Step 01: Selection of 600 SURF Feature Points

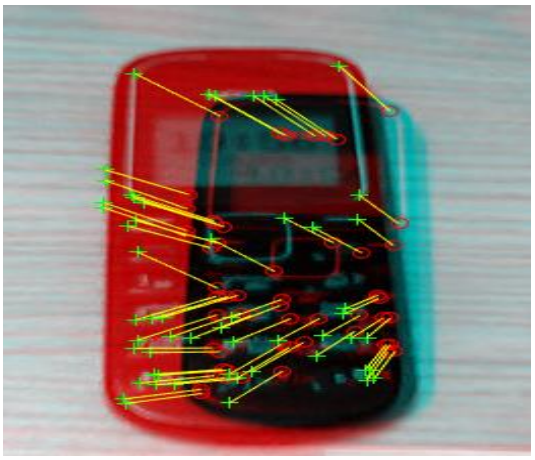


Figure 3.7.4.2.2: Step 03: Feature Matching by using KNN Algorithm

When features have been extracted from all n images, they should be matched. Since different images may view the same point in the world, by utilizing the k-d tree algorithm each feature is matched to k nearest neighbors. (For our case we select k = 4) [2]

3.7.5 Comparison of SIFT & SURF Features.

The efficiency and accuracy of the matching algorithm was compiled into two bar graphs depicting the successful match percentage for each algorithm, and number of correct pattern extractions in the database, respectively. Figure depicts the first graph that depicts the efficiency of each algorithm, which was measured by the number of correct matches of the Harbor Seal over the number of matching attempts. [25]

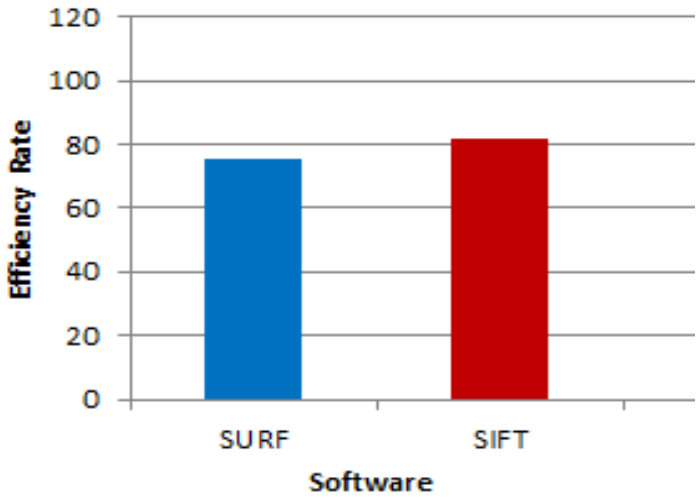


Figure 3.7.5.1: Efficiency of SIFT and SURF for Feature Matching

The x-axis represents the Algorithm, while the y-axis depicts the successful match percentage. The SIFT algorithm was the most successful algorithm in terms of efficiency in matching the database, with a match rate of 82%. SURF matched at an approximately 78.4% matching rate when compared with the same database as SIFT. Moreover, the ORB algorithm came in last with a successful matching rate of only about 54.4% under the same database SIFT and SURF used. Moreover, the I3s Software that was tested only matched with a rate of 38.8% when compared with the same database. The hypothesis is affirmed based on the results of this graph.

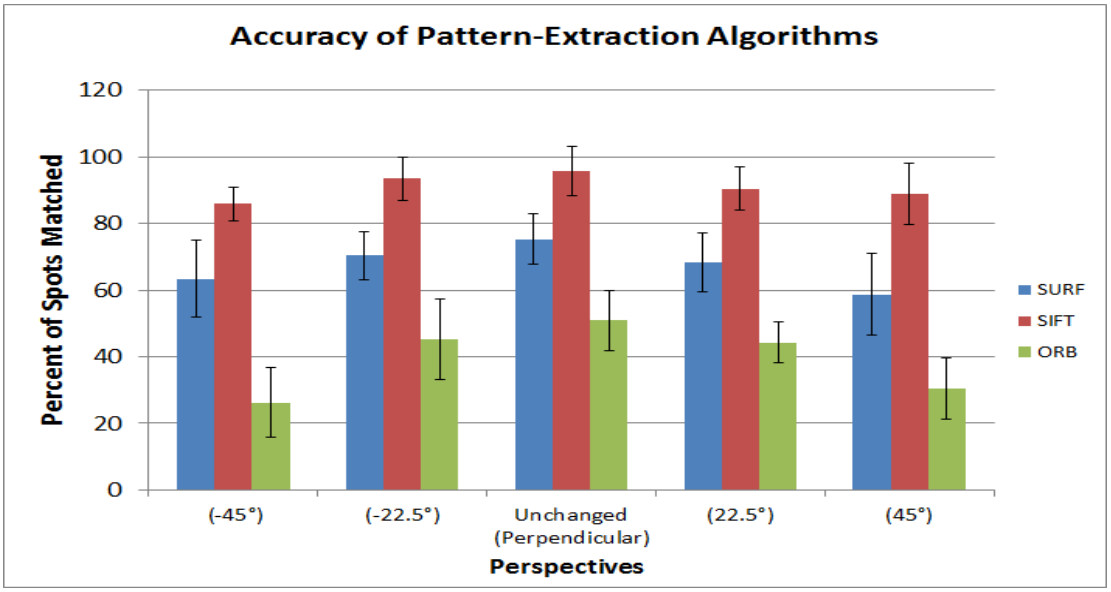


Figure 3.7.5.2: Accuracy of SIFT and SURF for Feature Extraction [26]

For the Accuracy of the Pattern Extraction Algorithm, which was measured by the number of mismarked points within each perspective group, the x-axis was the Seal-identification within each image set, and the y-axis is the number of correct pattern extractions. Likewise, the SIFT algorithm had the most amount of correct pattern extractions in each perspective, as it nearly outscored the other two formulas in every identification group. SURF and ORB follow the same general trend, with SURF ranking second in most of the identification groups, and ORB ranking last in every identification group in terms of the number of pattern extractions in each Seal ID. The hypothesis is supported that the SIFT operator is the more efficient software with SURF and ORB coming in second and third, respectively.

Biometric Analysis FAR (False Acceptance Rate) and FRR (False Rejection Rate) were then tested to see which program was efficient and significant in a field study. FAR is the number of false acceptances over the number of comparison attempts, and FRR is the number of false rejections over the number of comparison attempts. Based on David Lowe's paper, the value to be deemed significant and efficient was less than .05, or 5%. In both FAR and FRR, only the SIFT Developed program was deemed efficient and significant, with values less than .05 for both FAR and FRR.

As the SIFT has high accuracy of extracting the features as compared to SURF Features but the efficiency of SIFT and SURF algorithms for matching is almost same. SURF has another

advantage in that it take less computational power and processing time as compared to SIFT as shown in figure below [ref]

Algorithm	Feature matching Time
SIFT (Scale Invariant Feature Transform)	1.543 s
SURF (Speed Up Robust Feature)	0.546 s

Table 3.7.5.1: Processing Time of SIFT and SURF for Feature Matching [26]

Its means that SIFT requires three times more processing time as compared to SURF, so, we choose SURF feature extraction method for our case.

In MATLAB software we set a metric thresh hold = 600 which means we detect 600 interest point in all the images. The reason for choosing 600 interest points are that the time taken by the processor is lowest at 600 interest points as shown in figure below. If we choose less than or greater than 600 interest points than the processor will require high computational power as well as the processing time.

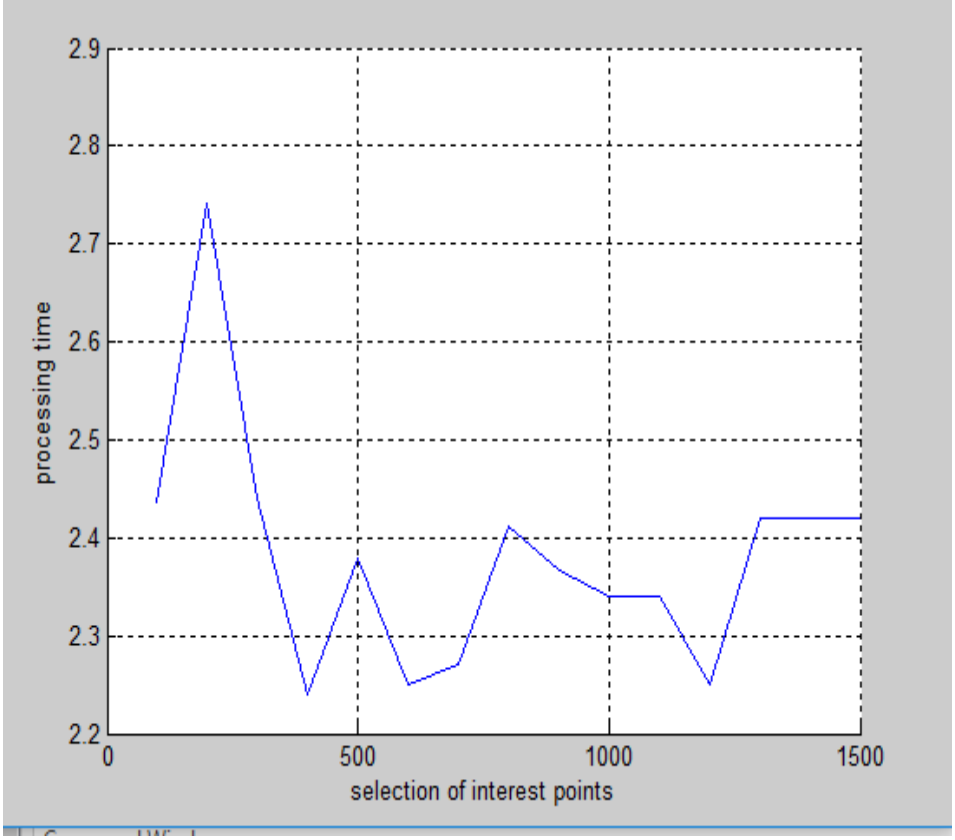


Figure 3.7.5.3: Graph between Selection of interest Points and processing time

The above graph shows clearly that the processing time at 400 Points and 600 points is low. At 400 the processing time is about 2.24 seconds while at 600 we have processing time of 2.25 seconds.

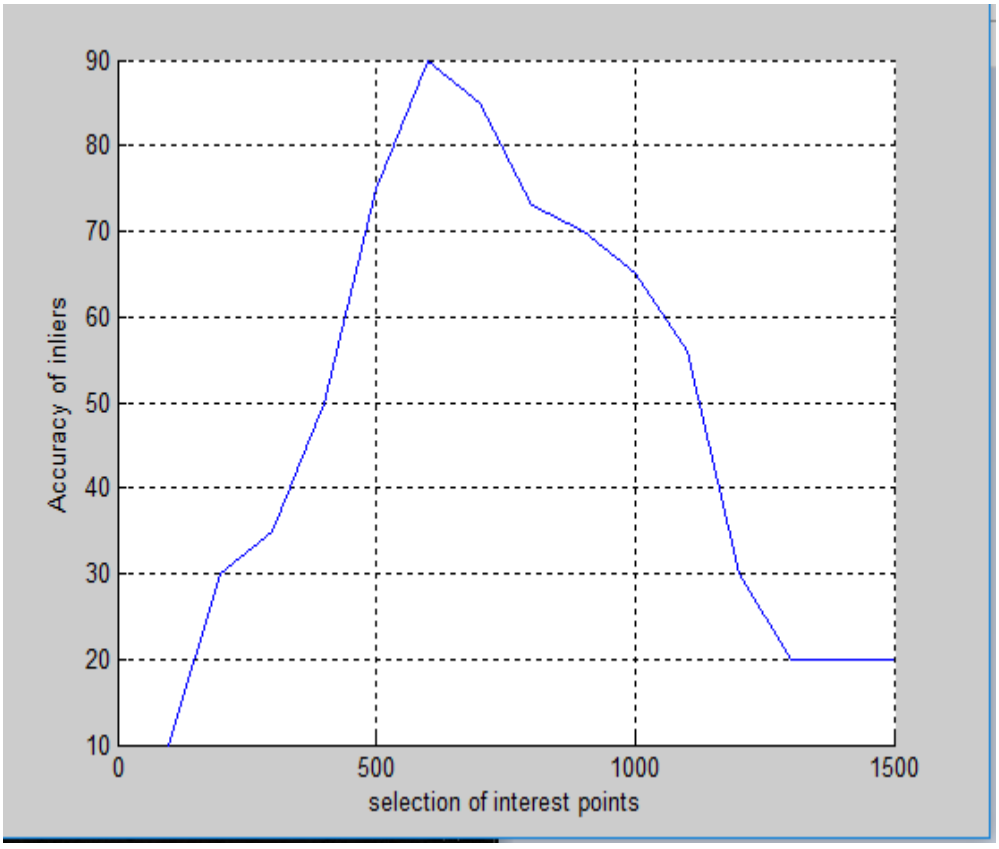


Figure 3.7.5.4: Graph between Selection of interest Points and Accuracy of Correct matches

As the graph clearly shows that accuracy at 600 is about 90% while accuracy at 400 is about 70%. So we select 600 interest points for feature extraction.

As the processing time at 600 interest points is low and accuracy of correct matches at these points is high so we extract 600 feature points from all the input images.

When features have been extracted from all n images, they should be matched. Since different images may view the same point in the world, by utilizing the k-d tree algorithm each feature is matched to k nearest neighbors. (For our case we select k = 4) [2].

3.8 Rejection of Outliers Features

In correct/ noisy matched features are removed by the rejecting the outlier features. It has been found that comparing the distance of potential match and best in correct match is effective strategy for rejecting outliers. So it is used to eliminate matches that correspond to

points that do not lie on an epipolar line. For this we set a epipolar threshold = 0.003 it means that any matched point that is has distance greater than thresh hold is not on the epipolar line, then it can be safely discarded as an outlier. Mathematically,

$$\frac{E_{match}}{E_{outlier}} > 0.003$$

Where E_{match} is, the distance of potential correct match and $E_{outlier}$ is the match distance as it is best matching outlier.

3.9 Image Matching

At this stage, our goal is to find all matching images that have common subset of 3D points that point will late become a 3D model.

From the feature matching, we observed that large number of feature matching between all images. Since each image could potentially match every other one, this problem appears in number of images. However, we have found it necessary to match each image only to a small number of neighboring images in order to get good solutions for the camera positions. We consider four images that have greatest number of feature matches to the current image. There are 7 parameters for our camera. These parameters are as follows,

- Rotation vector R
    - Rotation about x axis
    - Rotation about y axis
    - Rotation about z axis
- Translation Vector
    - Translation about x axis
    - Translation about y axis
    - Translation about z axis
- Calibration Matrix

3.10 3D Reconstruction

In the absence of a calibration grid, one view is clearly not sufficient to determine all the unknowns. Hence, we revisit here the intrinsic geometric relationship between two views

as shown in figure. In order to obtain the relationship between the two un calibrated views related by the displacement (R; T), consider the first the rigid body motion between two views:

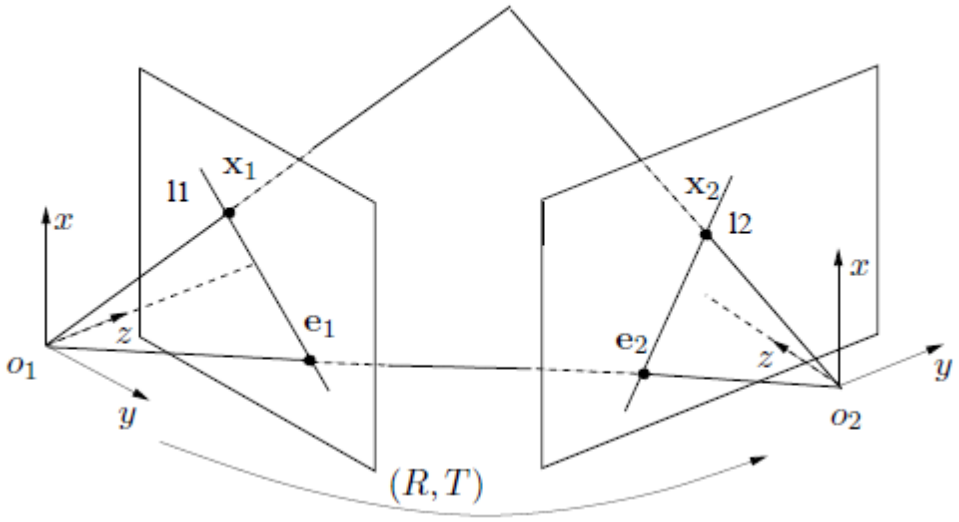


Figure 3.10.1: Relationship of Two Views [21]

$$F_{ij} = K_i^{-T} R_i \left[R_j^T t_j - R_i^T t_i \right] \times R_j^T K_j^{-1}$$

Where Fij is called Fundamental Matrix

Image matching involves strong estimation of the fundamental matrix Fij. Strong estimation of Fundamental Matrix means to create a strong relationship between all the views taken by the camera. For Strong estimation of F we use RANSAC method. It finds a set of inliers that have consistent epipolar geometry. If the number of RANSAC inliers $E_{inliers} > E_{match}$, the image match is declared. Where $E_{inliers}$ the minimum number of matches and $E_{matched}$ is a constant. All the inliers are then connected to each other to construct a 3D point cloud of the detected object.

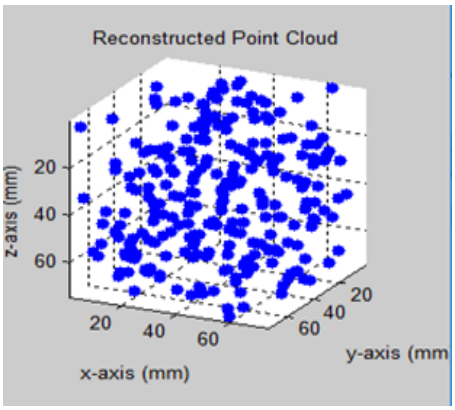


Figure 3.10.2: Result of 3D Reconstruction of object

Chapter 4: Object identification & Execution of best gripping features

4.1 Overview:

From the 3D reconstruction of the object as discussed in pervious chapter we can calculate the perimeters of the object such as length, width and depth of the object. In this chapter calculation of perimeters of the object, Human Hand Anthropometry and execution of best gripping features for the detected object is discussed in detail.

4.2 Object Identification:

Object identification is done by finding the perimeter so the object. The perimeters of the object can be calculated from the 3D reconstructed point cloud of the object.

For perimeters calculation is done by using distance formula as shown below

$$\boldsymbol{Distance} = \sqrt{(\boldsymbol{L_{max}} - \boldsymbol{L_{min}})^2 + (\boldsymbol{W_{max}} - \boldsymbol{W_{min}})^2 + (\boldsymbol{H_{max}} - \boldsymbol{H_{min}})^2}$$

Where,

$\boldsymbol{L_{max}}$ is the maximum value of matched featured in world co-ordinates along x axis.

$\boldsymbol{L_{min}}$ is the minimum value of matched featured in world co-ordinates along x axis.

$\boldsymbol{W_{max}}$ is the maximum value of matched featured in world co-ordinates along y axis.

$\boldsymbol{W_{min}}$ is the minimum value of matched featured in world co-ordinates along y axis.

$\boldsymbol{H_{max}}$ is the maximum value of matched featured in world co-ordinates along z axis.

$\boldsymbol{H_{min}}$ is the minimum value of matched featured in world co-ordinates along z axis.

4.2.1 Length of the object:

Length of the object is calculated by the following formula

$$\boldsymbol{Length\ of\ the\ object} = \sqrt{(\boldsymbol{L_{max}} - \boldsymbol{L_{min}})^2 + (\boldsymbol{0})^2 + (\boldsymbol{0})^2}$$

$$\boldsymbol{Length\ of\ the\ object} = \sqrt{(\boldsymbol{L_{max}} - \boldsymbol{L_{min}})^2}$$

$$\boldsymbol{Length\ of\ the\ object} = \boldsymbol{L_{max}} - \boldsymbol{L_{min}}$$

4.2.2 Width of the object:

Width of the object is calculated by the following formula

$$\boldsymbol{Width\ of\ the\ object} = \sqrt{(\mathbf{0})^2 + (\boldsymbol{W_{max}} - \boldsymbol{W_{min}})^2 + (\mathbf{0})^2}$$

$$\boldsymbol{Width\ of\ the\ object} = \sqrt{(\boldsymbol{W_{max}} - \boldsymbol{W_{min}})^2}$$

$$\boldsymbol{Width\ of\ the\ object} = \boldsymbol{W_{max}} - \boldsymbol{W_{min}}$$

4.2.3 Depth of the object:

Height of the object is calculated by the following formula

$$\boldsymbol{Height\ of\ the\ object} = \sqrt{(\mathbf{0})^2 + (\mathbf{0})^2 + (\boldsymbol{H_{max}} - \boldsymbol{H_{min}})^2}$$

$$\boldsymbol{Height\ of\ the\ object} = \sqrt{(\boldsymbol{H_{max}} - \boldsymbol{H_{min}})^2}$$

$$\boldsymbol{Height\ of\ the\ object} = \boldsymbol{H_{max}} - \boldsymbol{H_{min}}$$

4.3 Execution of Best Gripping Feature:

Execution of Best Gripping Feature is decided on the base of thresh holding function. Thresh holding function is set after literature and by doing no. of experimentations on different peoples hand.

4.3.1 Human Hand Anthropometry

Human Hand Anthropometry is shown in table given below [27]

4.3.1.1 Length of the Human Hand

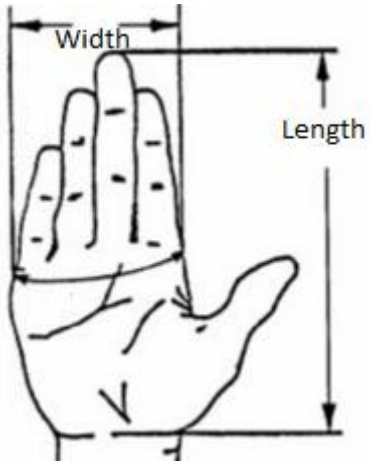


Figure 4.3.1.1.1: Human Hand Anthropometry [27]

Length of the Human hand is measure from the base of the hand at the wrist crease to tip of the middle finger. Length of the human hand (men and women) is shown in table 4.3.1.1.1.

Sr. #	sample	Percentile				
		1st	5th	50th	95th	99th
1.	Men	173mm	179mm	193mm	211mm	220mm
2.	Women	159mm	165mm	180mm	197mm	205mm

Table 4.3.1.1.1: Length of Human Hand

4.3.1.2 Breadth/Width of the Human Hand

The Breadth of the Human Hand is measured across the ends of metacarpal bones. Breadth of the human hand (men and women) is shown in table 4.3.1.1.1.

Sr. #	sample	Percentile				
		1st	5th	50th	95th	99th
1.	Men	81mm	84mm	90mm	98mm	100mm

2.	**Women**	71mm	73mm	79mm	86mm	89mm

Table 4.3.1.1.1: Breadth of Human Hand

The Human Hand Anthropometry shows that the human hand can grip any object whose length is 220mm or less so for best execution of gripping feature like human do, many xperiments are performed on 10 different peoples to check when human executes different grasping features from grasping different objects. From experimentation we observed the following

- Whenever the width of object is about 90mm to 150mm and the height of the object is about 25mm to 60mm, a human executes power grip to grip the object.
- A human hand also execute power grip when the length and width of object is about 50mm to 90mm and height is greater than 150mm.
- Whenever the width of object is about 10mm to 75mm and the height of the object is about 15mm to 25mm, a human executes tripod grip to grip the object.
- A human hand also execute pinch grip when width of object is about 10mm to 30mm and height is greater than 15mm to 25mm.

Chapter 5: 3D Model Based Validation

5.1 Overview:

After the detailed discussion on human hand anthropometry and execution of best gripping feature in pervious chapter, we performed 3D model based validation. In this chapter we will discuss how to create a 3D model, creating a dialog box to rotate 3D object at any angle and capture images at different angles, 3D object reconstruction and execution of best gripping feature for detected object in detail.

5.2 Steps of 3D Model Based Validation:

Following are steps of 3D model based Validation

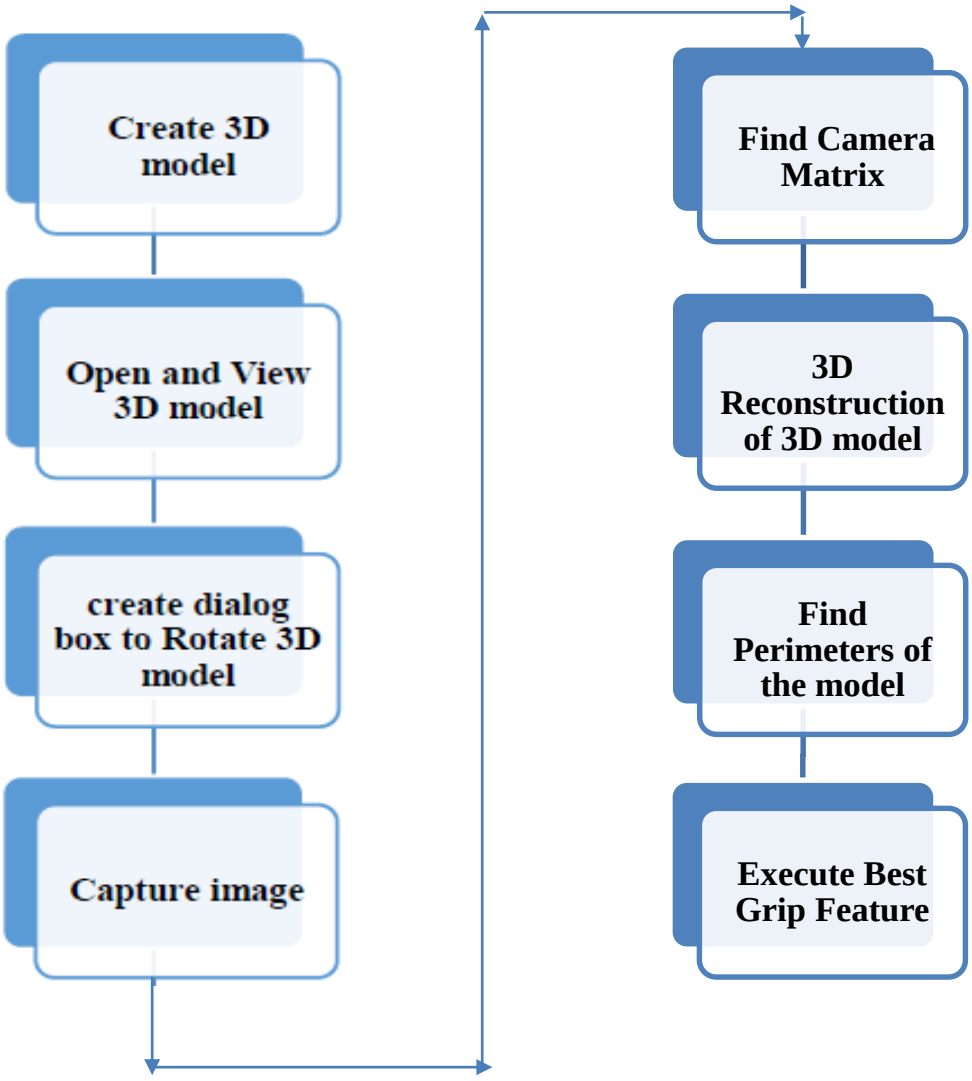


Figure 5.2.1: Steps of 3D model based Validation

For 3D Model Based Validation the first step is to create a 3D model by using 3D world Editor or by using Mesh Lab Software. An example of 3D world model is shown in figure below.

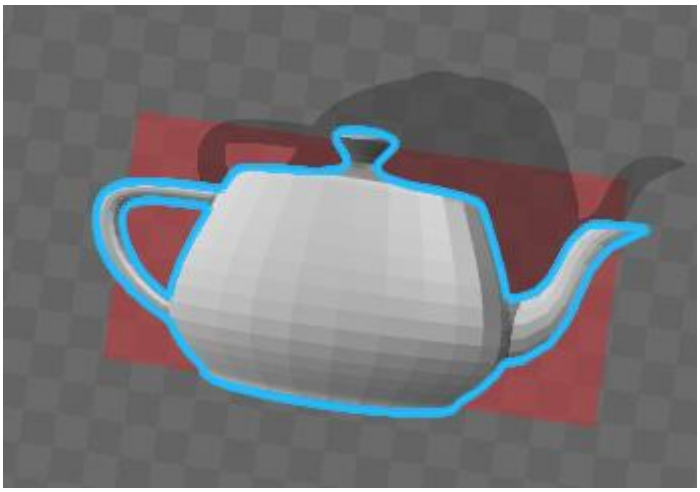


Figure 5.2.2: 3D Model created in 3D world Editor

After creating a 3D model the next step is to open and view the 3D model. The 3D world can be viewed by virtual Viewer, MATLAB offers two ways to view 3D world

- Internal viewer (the default method) and
- External viewer (integrated with your Web ser)

We use Internal Viewer method to show 3D model as shown in figure below

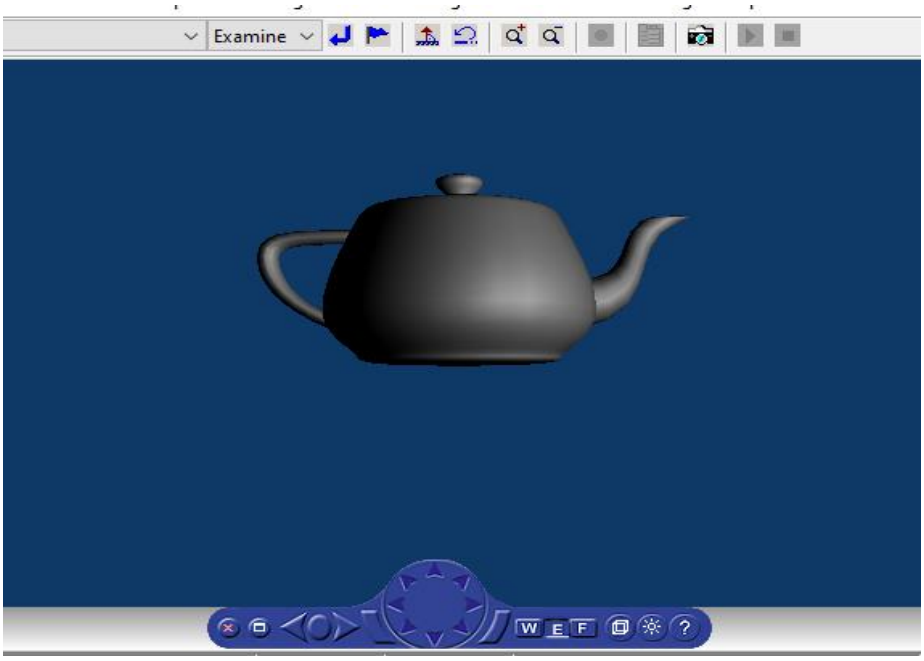


Figure 5.2.3: View of 3D Model

The dialog is used to interactively change field values of the VRML node referred to by the VRNODE object just created. This Dialog box is used to Rotate the 3D world at any angle about x axis, y axis, z axis. So our next step is to create a dialog box that rotate the 3D model from 0 to 360 degree about any axis. Dialog box is shown in figure below

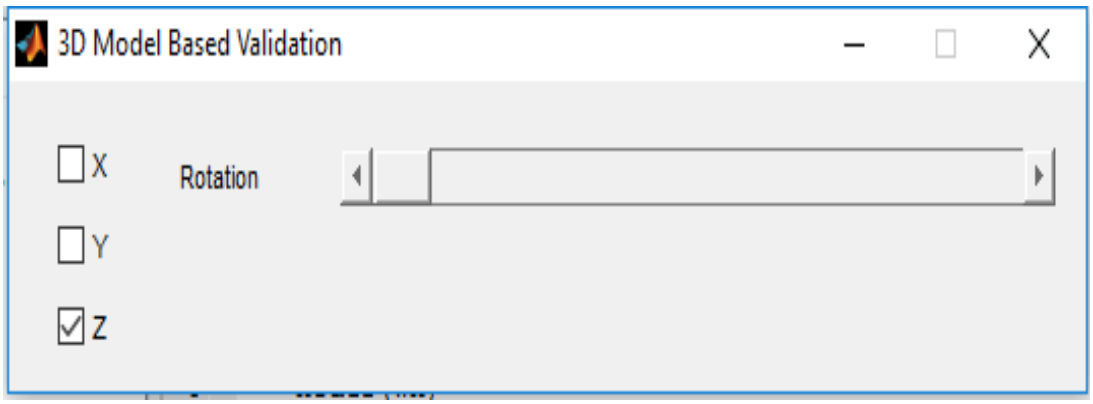


Figure 5.2.4: Dialog Box for Rotation of 3D Model

In dialog box we have a slide bar that rotates the model from 0 to 360 degrees. We also have 3 check boxes name x, y and z.

- If we check x box then 3D world only rotates about x axis
- If we check y box then 3D world only rotates about y axis
- If we check z box then 3D world only rotates about z axis
- If we check all boxes then 3d world rotates about all 3 axis.

After creating dialog box the next step is to capture the image at any desire angle and any desire axis of the 3D model by using capture command i.e built in MATLAB. First image is captured when rotation angle about x, y and z axis =0rad (as we do in engineering drawing for front view). 2nd image is captured at an angle greater than 0 and less than pi/2 rad about y axis (for side view).

After capturing images we can find out the camera matrices in order to find 3D geometry to estimate the world points. Algorithm for camera matrix is shown below

IF rotation is only about x axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} 1 & 0 & 0 \\ 0 & cosd(rot\ angle) & -sind(rot\ angle) \\ 0 & sind(rot\ angle & cosd(rot\ angle) \end{matrix}$$

END IF

IF rotation is only about y axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} cosd(rot\ angle) & 0 & sind(rot\ angle) \\ 0 & 1 & 0 \\ -sind(rot\ angle) & 0 & cosd(rot\ angle) \end{matrix}$$

END IF

IF rotation is only about z axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} cosd(rot\ angle) & sind(rot\ angle) & 0 \\ -sind(rot\ angle) & cosd(rot\ angle) & 0 \\ 0 & 0 & 1 \end{matrix}$$

END IF

IF rotation is about x and z axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} 1 & 0 & 0 \\ 0 & cosd(rot\ angle) & -sind(rot\ angle) \\ 0 & sind(rot\ angle & cosd(rot\ angle) \end{matrix} * \begin{matrix} cosd(rot\ angle) & sind(rot\ angle) & 0 \\ -sind(rot\ angle) & cosd(rot\ angle) & 0 \\ 0 & 0 & 1 \end{matrix}$$

END IF

IF rotation is about y and z axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} cosd(rot\ angle) & 0 & sind(rot\ angle) \\ 0 & 1 & 0 \\ -sind(rot\ angle) & 0 & cosd(rot\ angle) \end{matrix} * \begin{matrix} cosd(rot\ angle) & sind(rot\ angle) & 0 \\ -sind(rot\ angle) & cosd(rot\ angle) & 0 \\ 0 & 0 & 1 \end{matrix}$$

END IF

IF rotation is about x and y axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} 1 & 0 & 0 \\ 0 & cosd(rot\ angle) & -sind(rot\ angle) \\ 0 & sind(rot\ angle & cosd(rot\ angle) \end{matrix} * \begin{matrix} cosd(rot\ angle) & 0 & sind(rot\ angle) \\ 0 & 1 & 0 \\ -sind(rot\ angle) & 0 & cosd(rot\ angle) \end{matrix}$$

END IF

IF rotation is about x, y and z axis

$$\text{Camera Matrix} = \text{perspective transformation} * \begin{matrix} 1 & 0 & 0 \\ 0 & cosd(rot\ angle) & -sind(rot\ angle) \\ 0 & sind(rot\ angle & cosd(rot\ angle) \end{matrix} * \begin{matrix} cosd(rot\ angle) & 0 & sind(rot\ angle) \\ 0 & 1 & 0 \\ -sind(rot\ angle) & 0 & cosd(rot\ angle) \end{matrix} * \begin{matrix} cosd(rot\ angle) & sind(rot\ angle) & 0 \\ -sind(rot\ angle) & cosd(rot\ angle) & 0 \\ 0 & 0 & 1 \end{matrix}$$

END IF

5.2.1 Results of 3D model based Validation

Model 01

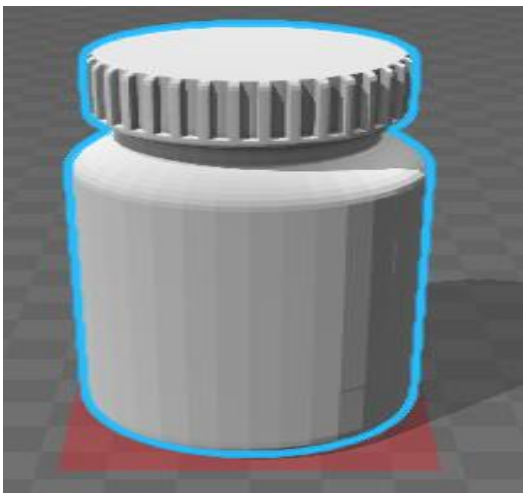


Figure 5.2.1.1: 3D Model

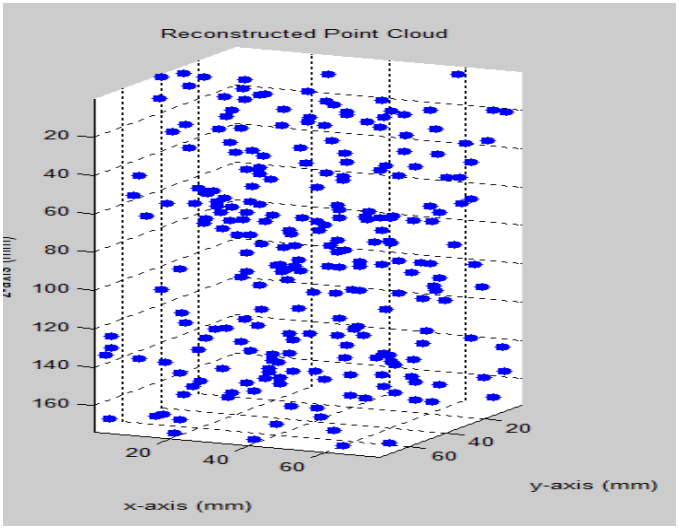


Figure 5.2.1.2: 3D Reconstruction of 3D Model

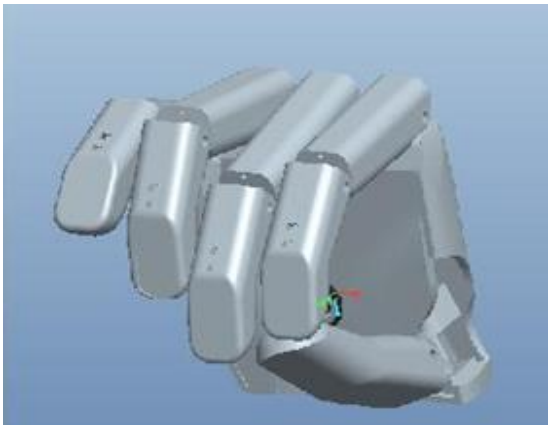


Figure 5.2.1.3: Best Gripping feature for grasping of 3D Model

Model 02

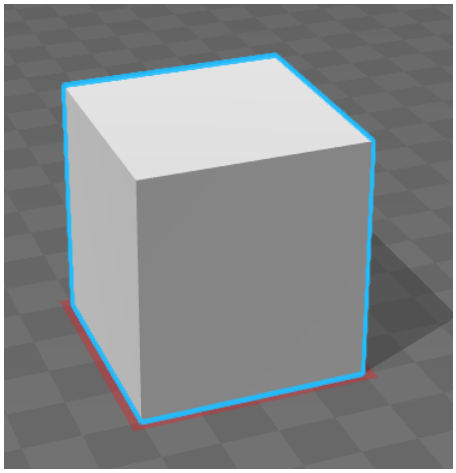


Figure 5.2.1.4: 3D Model

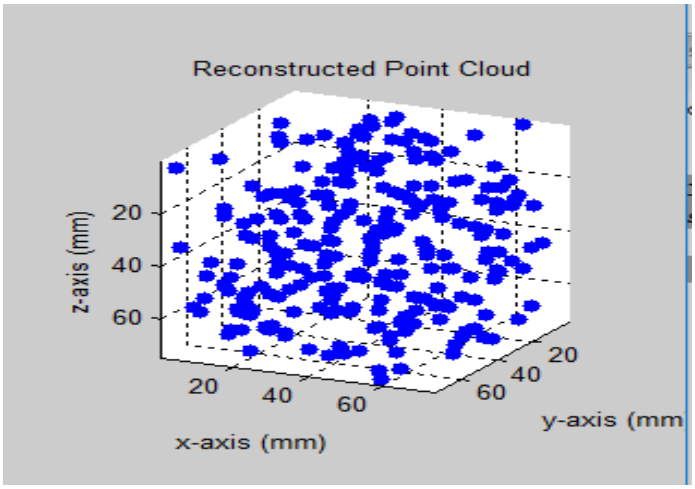


Figure 5.2.1.5: 3D Reconstruction of 3D Model

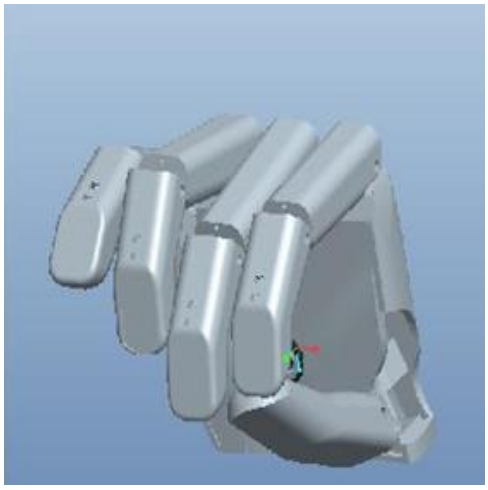


Figure 5.2.1.6: Best Gripping feature for grasping of 3D Model

Model 03

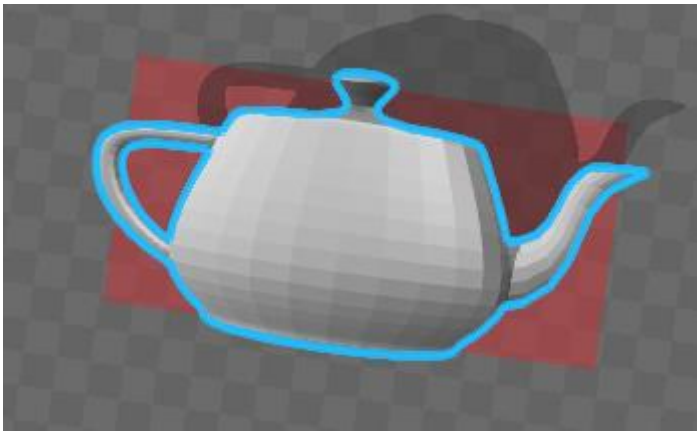


Figure 5.2.1.7: 3D Model

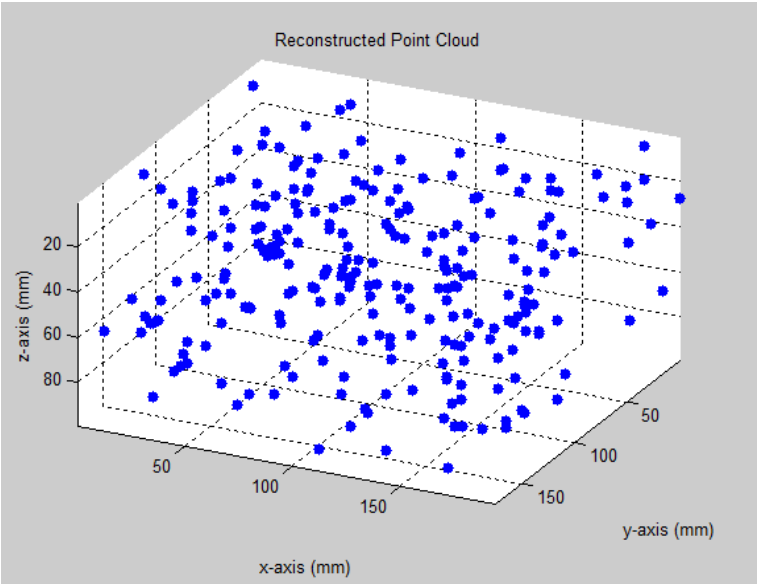


Figure 5.2.1.8: 3D Reconstruction of 3D Model

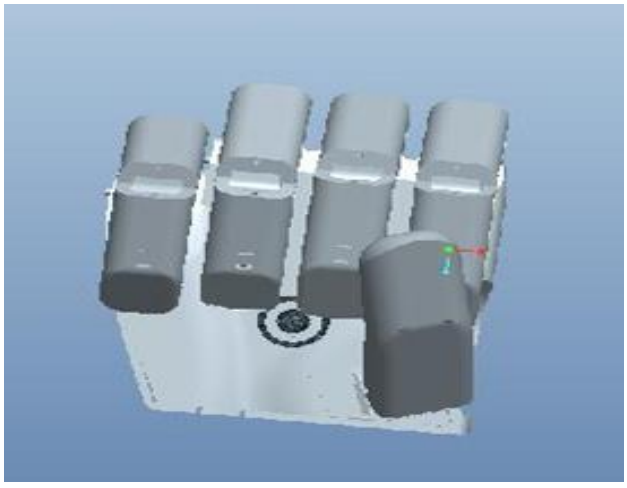


Figure 5.2.1.9: Best Gripping pattern for 3D model

5.2.2 Result

Exp. No.	Perimeters	No. of trials	Actual perimeters the of Object (mm)	Average Software Calculated Perimeters (mm)	Output	% Error
1	**L**	25	50	48	Tripod Grip	4%
	W		50	47		6%
	H		125	120.8		3.36%
2	**L**	25	50	47	Tripod Grip	6%
	W		50	47.6		5%
	H		50	46.8		6.4%
3	**L**	25	150	145	Power Grip	3%
	W		150	145		3%
	H		80	77		4%

Table 2.2.2.1: Result comparison of 3D model with software calculated values

Chapter 6: Experimental Results and Comparison with Actual Perimeters of the object

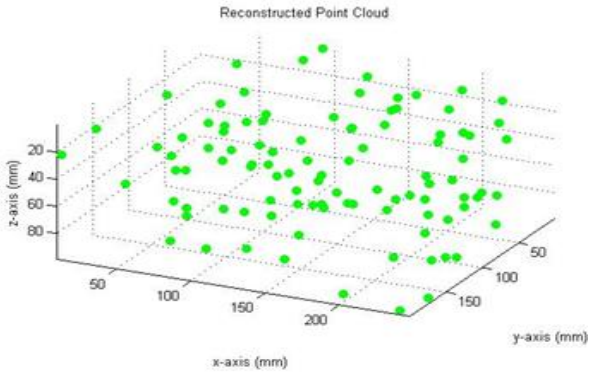
6.1 Overview:

In this chapter experimental results of different objects are shown, comparison of experimental results along with the software calculated results are compared and execution of best gripping pattern are shown in the chapter.

6.2 Results

Experiment 01:

(a)

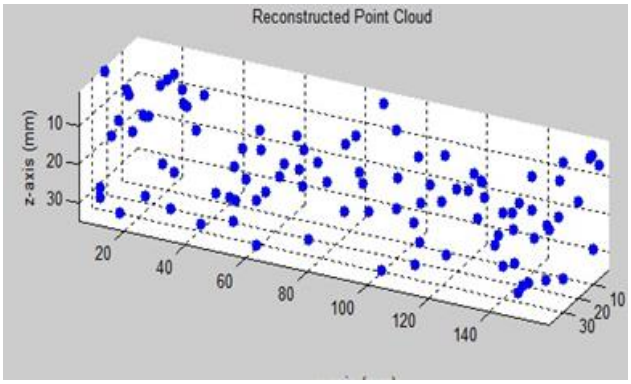


(b)

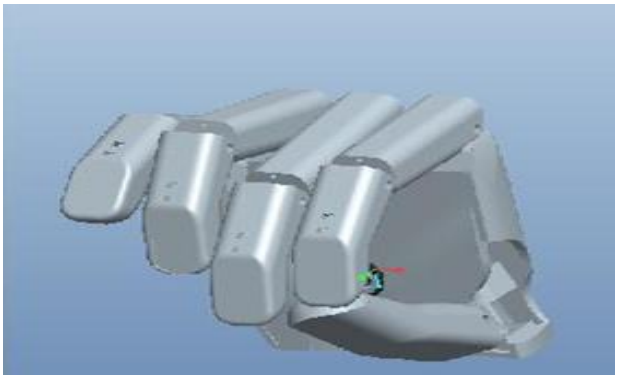
(c)

Figure 6.2.1: (a) shows the input images taken from the camera (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object

Experiment 02:



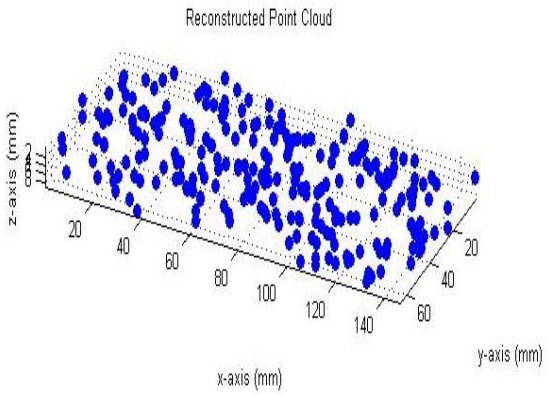
(a) (b)



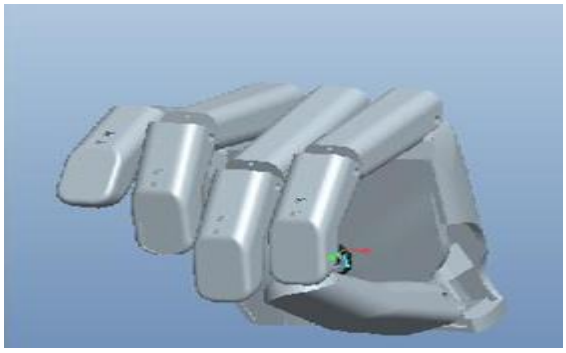
(c)

Figure 6.2.2: (a) shows the input images taken from the camera at different angles, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object

Experiment 03:



(a) (b)

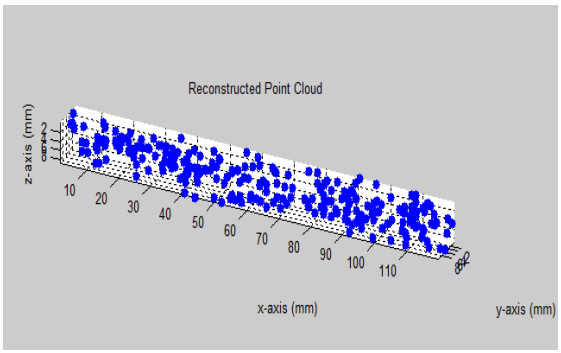


(c)

Figure 6.2.3: (a) shows the input images taken from the camera at different angles, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object

Experiment 04:

(a)

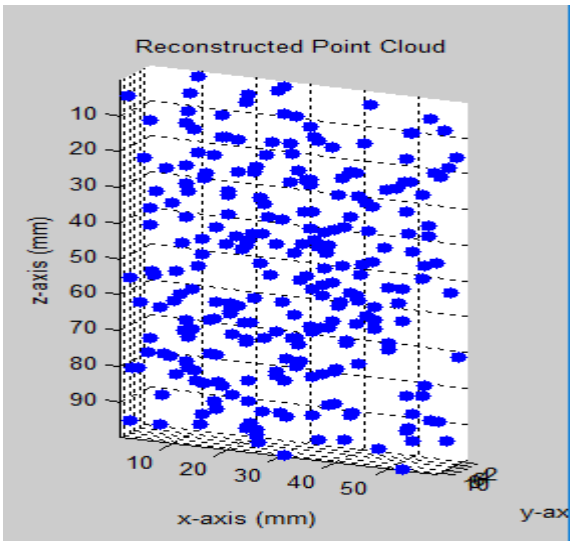


(b)

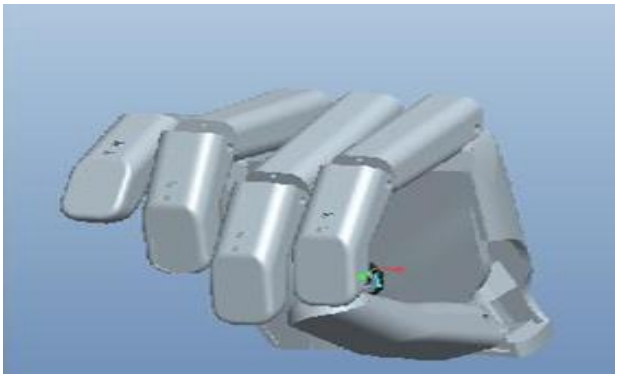
(c)

Figure 6.2.4: (a) shows the input images taken from the camera at different angles, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object

Experiment 05:



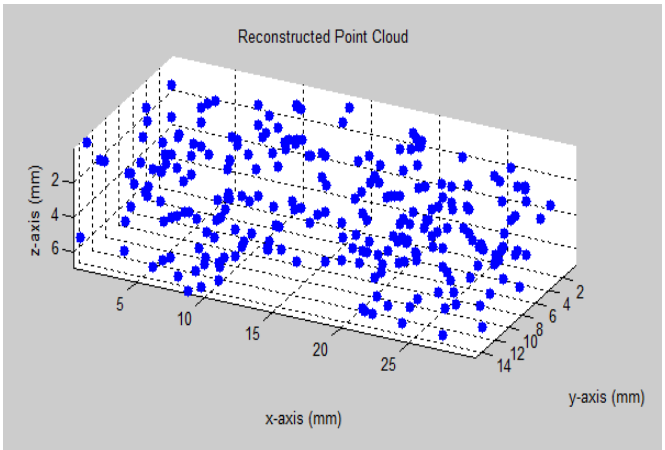
(a) (b)



(c)

Figure 6.2.5: (a) shows the input images taken from the camera, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected object

Experiment 06:



(a) (b)

(c)

Figure 6.2.6: (a) shows the input images taken from the camera, (b) shows the 3D reconstruction of the object, (c) shows the best grasping feature for the griping the detected

Exp. No.	Perimeters	No. of trials	Actual perimeters the of Object (mm)	Average Software Calculated Perimeters (mm)	Output	% Error
1	L	25	128	120.6	Power Grip	6%
	W		122	118.5		3%
	H		50	45.7		8.6%
2	L	25	156	151.6	Tripod Grip	3%
	W		41	36.05		12%
	H		38	33.8		10.5%
3	L	25	150	145	Tripod Grip	3%
	W		74	64		13%
	H		10	9		10%
4	L	25	135	123	Pinch Grip	9%
	W		10	8.9		11%
	H		10	9.7		3%
5	L	25	60	57.9	Tripod Grip	4%
	W		10	11.7		17%
	H		100	97		3%
6	L	25	30	29	Pinch Grip	3.33%
	W		15	13.7		8.6%
	H		8	6.8		15%

Table 3.2.1: Comparison between Actual and Software calculated perimeters

Chapter 7: Conclusion & Future Work

7.1 Comparative Study & Contributions

	Object 3D Reconstruction	Detect objects of all colours	Grip Features
IRIS Hand (2014)	No	No	Pinch, Tripod, Power
Sexana (2014)	No	Yes	Pinch Grip
VIPeR Arm (2015)	No	No	Pinch, Tripod, Power
Saqib (2016)	Yes	Yes	Pinch, Tripod, Power

Table 7.1.1: Comparison of our Research with other Recent Research

6.3 Conclusion:

A system that automatically detect the object and Construct a 3D scene from multiple input images taken from the camera embedded in the palm of hand. From 3D Reconstruction we can determine the perimeters of the object i.e. Length, Width, Height/depth the user is reaching for. The only limitation is that the user will only need to reach for an object and tell the device when to close. It executes pinch grip, power grip and tripod grip for grasping complex shape objects.

7.3 Future Work

As our Research is limited to only three grasping features pinch grip, tripod grip and Power Grip, so in future we can upgrade our system to more gripping pattern for grasping of all kind of objects as human do, we will also reduce processing time for object recognition and 3D reconstruction of object.

References

[1] [1] Sri Balaji V. Vellore, Samson Nivins, Vellore, Aravind, ***Upper Limb Prosthesis Using EMG Signal: Review***, The International Journal of Technology, vol. 2, issue 3, march 2015.

[2] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simpe Features.," in *International Conference on computer Vision and Pattern Recognition*, 2001.

[3] D. TR, L. Pezzin and E. Mackenzie, "Limb deficiency and emputation-epidemiology and recent trends in the US," South Med J, 2002.

[4] R. Szeliski and S. Kang, "Recovering 3D shape and Motion from Image Streams using Non-Linear Least Squares," Technical Repoort CRL 93/3, Digital Equipment Corporation, Cambridge Research Laboratory, 1993.

[5] C. Schmid and R. Mohr, "Local Gray Value Invariants for Image Retrieval," in *IEEE Transactions on Patteren Analysis and Machine Intelligence*.

[6] F. Schafflitzky and A. Zisserman, "Multi View Matching for unodered Image Set," in *European Conference on Computer Vision*, 2002.

[7] F. Rothganger, S. Lazebnik, C. Schmid and J. Ponce, "3D object Modelling and Recognition using Affine-Invariant Patches and Multi View Spatial Constraints," in *Internal Conference on Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

[8] R. V. Preedy, Handbook of Anthropometry: Physical Measures of Human Form in Health and Disease, Springer-Verlag New York, 2012.

[9] M. Pollefeys, "3D Modelling from Images," in *Europeanl Conference on Computer Vision*, Dublin, june, 2000.

[10] Niranjan and vedaant, "DEVELOPING A NON-INVASIVE WAY TO TRACK HARBOR SEALS USING PHOTO-IDENTIFICATION TECHNIQUES," google .

[11] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," in *European Conference on Computer Vision*, 2002.

[12] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Base line Stereo from Maximally Stable Extremal Regions," in *British Machine Vision Conference*, 2002.

[13] Y. Ma, S. soatto and S. Sastary, An Invention to 3D vision from Images to Model, 2001.

[14] D. Lowe, "Object Recognition from Local Scale-Invariant Features," in *International Conference on Computer Vision*, Corfu, Greece, September, 1999.

[15] D. G. Lowe, "Distinctive Image Features from Scale-Invariant keypoints," *International Journal of Computer Vision*, pp. 91-110, 2004.

[16] R. I. Hartley and A. Zisserman, Multi View Geometry in Computer Vision, Cambridge University Press, 2000.

[17] S. Casley, .. A. Jardo and D. Ozgoren, "IRIS Hand," WPI, 2014.

[18] M. Brown and D. Lowe, "Recognising Panoramas," in *9th International Conference on Computer Vision*, October, 2003.

[19] M. Brown, R. Szeliski and S. Winder, "Multi Image Matching using Multi Scale Orient Patches," in *International Conference on Computer Vision and Pattern Recognition*, San Diego, June, 2005.

[20] J. Beis and D. Lowe, "Shape Indexing using Approximate Nearest-Neighbor Search in High Dimensional Spaces," in *International Conference on Computer Vision and Pattern Recognition*, 1997.

[21] A. Baumberg, "Reliable Feature Matching Across Widely Separated Views," in *International Conference on Computer Vision and Pattern Recogntion*, 2000.

[22] "wiki pedia," [Online]. Available: https://en.wikipedia.org/wiki/Speeded_up_robust_features. [Accessed 2016].

[23] Robert Edwards, Kyle Lafontant, Nuttaworn Sujumnong, Jared Wormley, '*Vision Based Intelligent Prosthetic Robotic Hand*', WPI smart Robotics Lab, 2015

[24] A Saxena, J. Driemeyer, Andrew Y. Ng, *Robotic Grasping of Novel Objects*, The International Journal Robotics Research, 2014.

[25] Niranjan and vedaant, "*DEVELOPING A NON-INVASIVE WAY TO TRACK HARBOR SEALS USING PHOTO-DENTIFICATION TECHNIQUES*," google

[26] P M Panchal, S R Panchal, S K Shah '*A comparison of SIFT and SURF'* International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013

[27] https://palmreadingperspectives.wordpress.com/2011/05/29/hand-anthropometry-from-leonardo-da-vinci-to-nasa-us-army/

Appendix

Pseudo Codes :

1. 3D Reconstruction of the Object

```
Camera Calibration

//Calibration of camera is done by using Camera Calibration app of MATLAB Software

input: Image Acquisition of object from Different Angles

FOR all input images

   Convert RGB images to Binary Image

   Find Connected Regions of Binary Images

   IF region has < 1000 pixels

        Discard that region

   END IF

   Detect SURF Features from the input images

   Initiate KD Tree Algorithm for matching

   FOR every images

select m matching images with maximum no. of matching features with images

IF the distance between two matching point > 0.03

  Eliminate that matching point

        END IF

Find the translation, rotation of camera of 3D geometry to estimate the world points

   END FOR

END FOR

output: Reconstruction of 3D point Cloud
```

2. Execution of Best Gripping Features:

```
input: Data of 3D Reconstructed Point cloud
//All Dimensions are in millimetres
IF      Length < 220 & 90<Width<150 & 25<Height<60
   Execute Power Grip
END IF
IF Length < 220 & 30 < Width < 100 & 100 < Height < 250
   Execute Power Grip
END IF
IF Length < 220 & 20 < Width < 70 & 12 < Height < 35
   Execute Tripod Grip
END IF

IF Length < 100 & 10 < Width < 30 & 75 < Height < 125
   Execute Tripod Grip
END IF
IF Length < 220 & 10 < Width < 30 & 10 < Height <25
   Execute Pinch Grip
END IF

output: Best Grip Feature for Grasping the object
```