

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES

DEVELOPMENT OF A ROBOTIC ARM
CONTROLLER BY USING HAND GESTURE
RECOGNITION IN MATLAB ENVIRONMENT

by

İbrahim Baran ÇELİK

March, 2011

İZMİR

**DEVELOPMENT OF A ROBOTIC ARM
CONTROLLER BY USING HAND GESTURE
RECOGNITION IN MATLAB ENVIRONMENT**

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
In Electrical and Electronical Engineering**


**by
İbrahim Baran ÇELİK**

March, 2011


İZMİR

M.Sc THESIS EXAMINATION RESULT FORM


We have read the thesis entitled “**DEVELOPMENT OF A ROBOTIC ARM CONTROLLER BY USING HAND GESTURE RECOGNITION IN MATLAB ENVIRONMENT** ” completed by **İBRAHİM BARAN ÇELİK** under supervision of **ASSOC. PROF. DR. MEHMET KUNTALP** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.


.....
Assoc. Prof. Dr. Mehmet KUNTALP

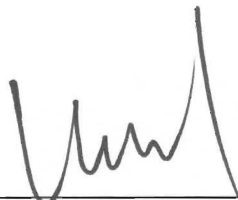
Supervisor


.....
Asst. Prof. Dr. Ahmet ÖZKURT

(Jury Member)


.....
Asst. Prof. Dr. Adil ALPKOÇAK

(Jury Member)


.....
Prof. Dr. Mustafa SABUNCU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to acknowledge the contributions of those people who supported me in the creation of this thesis:

Thanks to my supervisor, Assoc. Prof. Dr. Mehmet KUNTALP, who encouraged me to pursue my master's degree. You have taught me not only how to do research, but also how to be a real teacher with your unforgettable sentence "Teacher is not the one who opens the door for you, must be the one who shows the door that should be opened by you".

Thanks also to Asst. Prof. Dr. Ahmet ÖZKURT. Your advice, your insight and your guidance have been invaluable.

Thanks to all Dokuz Eylül Üniversitesi, Electric & Electronic Engineering Department for giving me the chance to face the real science and to know definition of a real engineer.

Thanks to my parents, İlmen and Mehmet Şükrü ÇELİK, who have encouraged and supported me in all that I have ever done.

Ibrahim Baran ÇELİK

DEVELOPMENT OF A ROBOTIC ARM CONTROLLER BY USING HAND GESTURE RECOGNITION IN MATLAB ENVIRONMENT

ABSTRACT

This thesis deals with the robotic arm controller using image processing in the field of Human-Machine Interaction (HMI). There are two different methods used to analyze to control the robotic arm, the main aim of them is getting the hand gesture information without using tool that helps the system to extract data easier (ex. glove or wrist band). After segmentation of the hand, the first method is comparing of all pre-stored data in the database at the Template Matching Algorithm, the second method is Signature Signal, distance signal between edge of the hand and center of hand, Signature Signal is used to find where the fingertips are and to count the number of them. The detailed test results and their conclusions show that both algorithms can be used for controlling after calibration. Both methods have enough calculation speed to be used in continuous frame capturing sequence.

Keywords: Robot controlling, computer vision, human machine interaction

MATLAB ORTAMINDA EL İŞARETİ TANIMLAMALI ROBOT KOL KONTROLCÜSÜ GELİŞTİRME

ÖZ

Bu tez İnsan-Makine Etkileşimi (İME) alanında olan görüntü işleme kullanılan robot kol kontrolü ile ilgilidir. Robot kolu kontrol etmede iki farklı analiz metodu kullanılmıştır, bu iki metodunda ana hedefi el sistemden daha kolay veri almada kullanılan herhangi bir araç kullanmadan el işareti bilgisini almaktır (ör. eldiven yada bilek bandı). Elin segmentasyonundan sonra ilk metod tüm önceden kaydedilmiş verilerle Şablon Eşleştirme algoritmasında karşılaştırmaktır. İkinci metod Sinyal İmzası algoritması temellidir, elin dış sınırları ile merkez noktası arasındaki mesafe temelli olan sinyaldir, Sinyal İmzası parmak uçlarının yerini bulmada ve onları saymada kullanılır. Detaylı test sonuçları ve sonuçların yorumları her iki algoritmanında robot kontrolünde kalibrasyon yapıldıktan sonra kullanılabilineceğini göstermektedir. Her iki metodta sürekli kamera veri işleme için yeterli hesaplama hızına sahiptir.

Anahtar Kelimeler : Robot kontrolü, bilgisayar görü sistemleri, insan makina etkileşimi

CONTENTS

	Page
THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ.....	v
CHAPTER ONE – INTRODUCTION	1
1.1 Problem Definition	1
1.2 Scope of the Thesis	1
1.3 Outline	2
CHAPTER TWO – GESTURE ANALYSIS AND ROBOT CONTROLLING ...	3
2.1 Introduction.....	3
2.2 Human Gesture Analysis	5
2.3 Robot Controlling	6
CHAPTER THREE –DIGITAL IMAGE ANALYSIS.....	8
3.1 Color Image Processing	8
3.1.1 Color Models.....	10
3.1.2 Color Models Conversions.....	11
3.2 Some Basic Relationship Between Pixels	13
3.3 Moment Based Image Analysis	14
3.4 Edge Detection.....	20
3.5 Zooming-Shrinking Digital Images	20
3.6 Image Rotating.....	21
3.7 Morphological Operation-Region Filling	22

CHAPTER FOUR -- ROBOTIC ARM	23
4.1 Properties of Robotic Arm	23
4.2 Properties of Parallax Servo Controller Card	24
CHAPTER FIVE -- SEGMENTATION OF HAND	25
5.1 Program Algorithm	25
5.2 Calibration	26
5.3 Video Processing	27
CHAPTER SIX -- HAND GESTURE RECOGNITION	33
6.1 Template Matching Algorithm	33
6.2 Signature Signal Algorithm	35
6.3 Performance Analysis of Algorithms	42
CHAPTER SEVEN -- FEATURE EXTRACTION.....	43
7.1 Local Feature Extraction.....	43
7.2 Global Feature Extraction.....	46
CHAPTER EIGHT -- CONCLUSION	48
8.1 Conclusion	48
8.2 Proposed Feature Work	49
APPENDICES.....	53
Appendix A: Application of software.....	53
Appendix B: Specification of servo motor controller.....	54
Appendix C: Test Results	55

CHAPTER ONE

INTRODUCTION

1.1 Problem Definition

Computer Vision based robot controlling has been a popular research topic in recent years and is an important application area for robotic systems. The most important advantage of this controlling is to avoid the unwanted and dangerous situations because the user has the advantage of changing the movements at that instance, for example for a researcher who make research about any kind of issue in an the isolated laboratory this kind of controlling is the best for everyone. Another advantage of that kind of controlling is that the user having a physical problem can use the robot in order it to help him/her to accomplish any tasks.

Although this type of controlling studies were done based on mechanical devices such as potentiometer of data gloves in previous years, recent studies related to the capabilities of computers and new technologies developed in the field of Computer Vision bring us a better performance and using hand itself without using any extra devices where robot observes the user visually.

In this thesis, for the system's robotic part, a handmade servo motor consisting of four axis robotic arm is used and for the controlling of the servos parallax motor driver card is used.

1.2 Scope of the Thesis

As for the system presented in this paper, the first method to control the robot arm is the Template Matching Algorithm while the second algorithm is the Signature Signal. These algorithms have different methods to recognize the hand gesture. There are two kinds of feature groups to extract from the hand based on the location of the hand. These features are Global and Local Features. Local Features give us the hand gesture information of the hand for the controlling of the gripper of robot-arm, and Global Features give us the hand's position in 3-D dimension.

The system is designed such that if the distance between the center of the hand and center of frame is small enough, Local Features are extracted; otherwise, Global Features are extracted. Template Matching's Local Features are found by matching the segmented hand with correct gesture in the data base. Signature Signal's Local Features are found by getting the code signal from the hand, finding the locations of finger in the signal and counting them.

1.3 Outline

Introduction is given in Chapter 1. Chapter 2 explains the concept of hand gesture analysis, robot controlling and previous research and applications about this subject.

This thesis contains many different Image Processing Techniques and algorithms. Details of them are given for background information in Chapter 3.

Robotic Hand and parallax Servo Motor ControllerCard is explained in Chapter 4.

As mentioned before, both algorithms contain same calculations until segmentation of the hand. Visual analysis for segmentation of hand is detailed step-by-step and the results are demonstrated in Chapter 5.

Although Gesture Recognition algorithm is under Local Feature extraction its considered to be better explaining it under distinct headers in Chapter 6, because it is the most important part of the thesis. Both methods (Template Matching & Signature Signal Algorithm) are explained in detail with their performance analysis.

Chapter 7 contains "Global" and "Local" feature extraction for robot controlling in detail and finally Chapter 8 concludes thesis.

CHAPTER TWO

GESTURE ANALYSIS AND ROBOT CONTROLLING

2.1 Introduction

Human-Machine Interaction (HMI) area is one of the most important research fields in development of successful robotic systems. Programming was the primary methods for interaction in the early years of robotics (Wu & Huang, 2001). As the computers capabilities are improved day by day one of the most used interaction type, mechanical type like giving command with mice will start to leave the new ones behind because these capabilities help us to develop new methods.

There are lots of new developed tools in HMI in recent years. Microsoft Kinect is one of them which is a special RGBD camera created for Microsoft's XBOX 360, to be used as a controller substitute and an extra input device for specific image processing and recognizing operations that exploit the use of the Kinect (Microsoft, 2011) because of this sensor has capabilities of having USB connection and giving the depth information of data.

There are so many developed projects with this tool, Chang et al. developed a Kinect-based rehabilitation system which assists therapists in their work with patients who has no motor abilities (Chang & Chen & Huang, 2011). The algorithm used the motion tracking data provided by the Kinect to check whether the patient's movements reached the rehabilitation standards. White et al. developed a virtual environment for stroke rehabilitation that tracks patients' arm movements during reaching exercises (White & Searleman & Carroll, 2005). Images are projected onto three walls of an enclosed space to simulate a kitchen setting with objects for the patient to reach for, and the patient's arm movement is tracked using a sensor attached to his or her arm. Rizzo and others at the University of Southern California studied video games that require player movement could motivate persons at risk for obesity to engage in physical activity (Rizzo & Lange & Suma & Bolas, 2011).

Hand gesture is one of the most important non-speech kinds of interaction. Gestures range from simple actions to more complex ones like emotions (Pavlovic & Sharma & Huang, 1997). Human arm tracking system is designed in (Bernardo & Goncalves & Perona, 1996). The arm is modeled as two truncated circular cones with spherical joints. The system works as follows: the arm image and stored image are compared and the joint is iteratively moved until matching occurs. Before starting the system, the position of the shoulder must be configured to the system manually.

Pointing based gestures are used in (Littmann & Drees & Ritter, 1996), where the system works as follows: the user points the objects that must be taken by robot in the environment then the robot arm takes that object. After taking the object, the user can put the objects by pointing the area where the robot should go. This work is ANN based. ANN is used to detect the pointing hand from the picture and the location where the hand points to. There are two cameras supplying the system. System accuracy is found to be 1 ± 0.4 cm in 50×50 cm environment.

Visual input of hand of user is analyzed in (Ude & Atkeson, 2001) by using color and shape information. The system determines the trajectory of the human hand and a humanoid robot follows the same trajectory in order to realize 10 different movements. Although the system operates successfully, the need for using stereo vision is its drawback.

A mobile robot is driven in (Rybski & Voyles, 1999) by hand gestures. The robot is able to detect commands like approach, grasp, follow etc... The gestures are captured and classified by using Hidden Markov Model. The robot can also do previously given commands before required time because the robot learns by gestures.

There is 3-D camera used in (Amit & Mataric, 2002). Whole body is learned by Hidden Markov Model and while body moves, a 3- D human simulator does the same movement in the computer.

There is humanoid robot is used in (Cheng & Kuniyoshi, 2000). This system can mimic the upper part of the human body, has the same 24 degree of freedom as the

human body upper part with two arms, head and torso. This work is based on color values aspect ratio and depth information. The system has auditory input which helps the robot to detect sound sources.

Gestures need dynamic and/or static configurations of the human hand for the Human-Machine Interaction. Hand, arm and sometimes body must be measured by the machine. In previous years this problem was tried to overcome by using some mechanical devices, which measure human hand or other places position, but the disadvantage of that method is that the user must carry heavy equipment, and this is far from being natural. The improved technology vision which is based on cameras and image processing techniques to recognize the necessary place is preferred. This kind of based method is preferred highly for HMI because it's also primary method of interaction among human (Pavlovic & Sharma & Huang, 1997).

2.2 Human Gesture Analysis

Using natural appearance of hand as a manipulator is desired when performing necessary tasks in HMI applications. In a taxonomy given in (Pavlovic & Sharma & Huang, 1997), hand/arm movements are classified as follows:

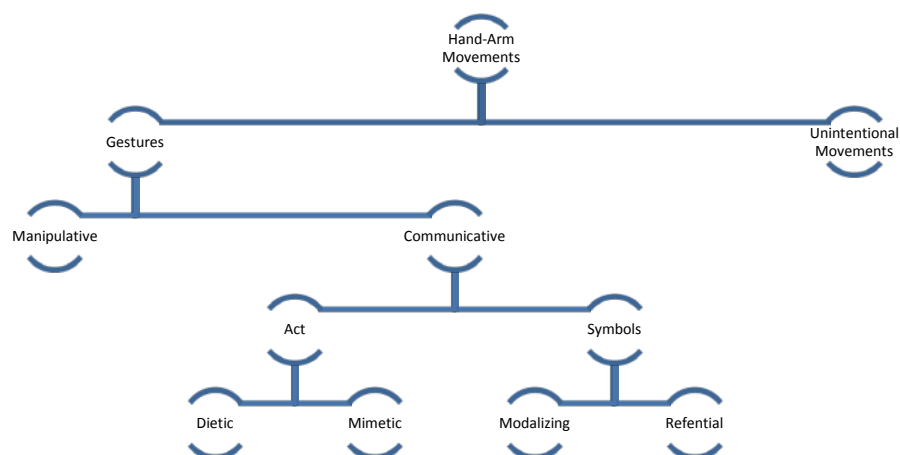


Figure 2.1 Taxonomy of gestures

Unintentional movements do not carry information. There are two sub classes in gestures Manipulative Gestures and Communicative Gestures. Manipulative gestures

are termed as affecting an object in real world, for example pushing a box from a place to another place. Communicative gestures has two sub classes, these are Acts and Symbols. Symbol gestures have linguistic role, they symbolize some reference action or they are used as moralizer. Acts gestures are directly related to the vitalization of the movement. Acts are either mimetic acts such as simulation of that act or deictic acts such as pointing acts.

Visual Gesture analysis consists of three stages:

1. Hand/Arm Localization and its segmentation: Area of interests (hand/arm) is extracted from all images. Most used ideas are skin color and motion information, this section is complex task because of consisting of image has thousands of unwanted items.
2. Hand/Arm Image Feature Extraction: The features those must be taken from the hand depends on the operation needs, for example if the operation is just related about the position of hand we don't need the fingers situations, but there are also kind of operations those needs to know fingers situation(like in this thesis).
3. Hand/Arm Model Parameter Calculation: Computed parameters in this step depends on application. Position of the hand is useful for a tracking system but for recognition system position of hand is not most wanted parameter.



Figure 2.2 Visual gesture analysis stage

2.3 Robot Controlling

The field of Biomedical Robotics has recently attracted many robotics research groups who developed a variety of solutions for the application of robotics in the different aspects of clinical activities such as the exploitation of physiological knowledge or laboratory explorations under dangerous conditions in the development of biomedical robots. The main areas that developed within this field are those of Robotics for Surgery, Rehabilitation Robotics, and Bio Robotics. Research of controlling those robots covers a large area of robotics, such as artificial intelligence, computer vision, pattern recognition and machine learning.

In order to control any robot with Visual Control System, information defining the movement to be controlled must be supplied to robotic system. Robot must obtain and evaluate that information by itself. Most natural way of controlling the robot may be done by visually with analyzing and extracting the necessary information's for the order of movement (Triesch & Von der Malsburg, 1998).

CHAPTER THREE

DIGITAL IMAGE ANALYSIS

3.1 Color Image Processing

Color is a powerful descriptor for simplifying object identification and extraction in automated image analysis (Gonzales & Woods, 2002).

Color image processing has two sub-areas:

- Full color (Full color sensor, TV camera etc)
- Pseudo color

All colors those detected by a Human-Being are combination of 3 called primary colors, Red(R) Green(G) Blue(B). Wavelengths are 700 nm, 546 nm, 435 nm. These three main colors cannot generate all spectrum colors by acting alone. Their wavelength must vary in order to generate any color. Secondary colors are the combination of primary colors.

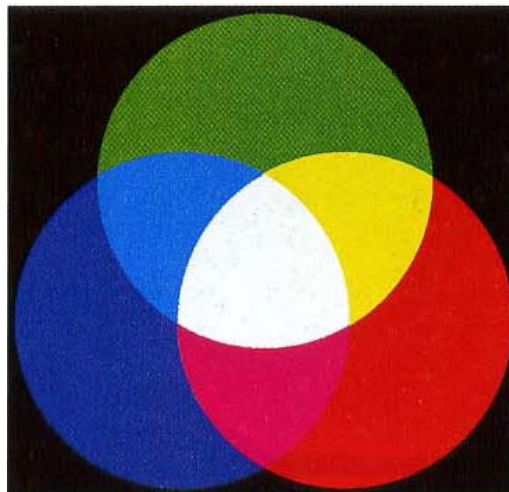


Figure 3.1 Primary and secondary colors of light and pigments (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 302-303).

There are three main characteristics used to distinguish one color from another

- Brightness, Chromatic notation of intensity

- Hue, it's the association of dominant wavelength of mixture of light waves. When an object is called Green ,its hue is specified
- Saturation, Refers to relative purity of amount of white light mixed in hue. Amount of white in the mixture has inverse ratio with saturation value.

Chromatics is the combination of Hue and Saturation, so a color can be characterized by its brightness and chromaticity.

Tristimulus values are the amount of R G B those are needed to form a color. They are denoted as X, Y and Z respectively. Color then specified by its trichromatic coefficients. Note that $x+y+z=1$;

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$

We can see the Chromatic diagram that shows colors on x-y plane. The point Green for example has approximately %60 green %25 Red and %13 Blue

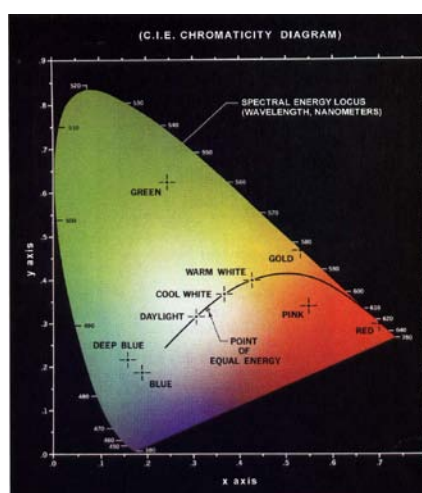


Figure 3.2 Chromaticity diagrams (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 288)

3.1.1 Color Models

Color models are designed and modeled for facilitation and specification of colors in some standar.RGB and HSI color models are used for this thesis (Gonzales & Woods, 2002).

3.1.1.1 RGB

Used mainly for color monitors and color video cameras

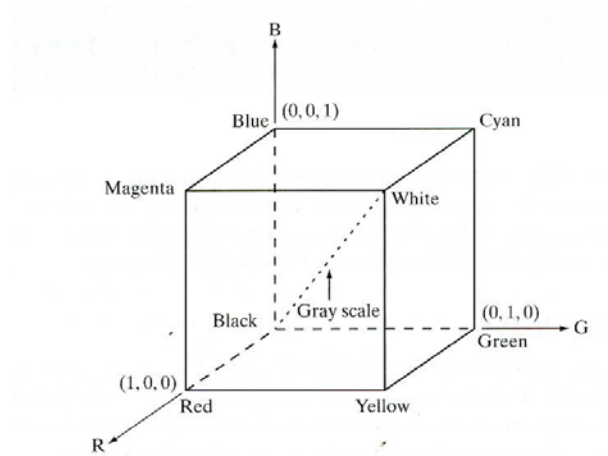


Figure 3.3 Schematic of the RGB color cube(Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 290).

3.1.1.2 HSI

I is intensity and decoupled from the color information in the image and S are related to the way in which human being perceive color. That makes HSI model ideal one for image processing algorithms.

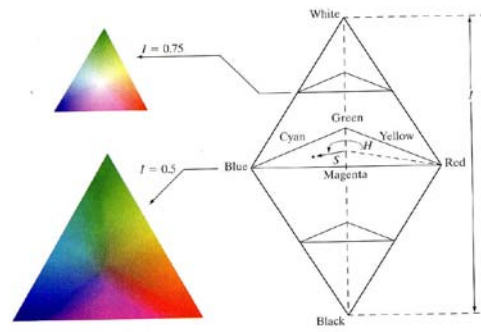


Figure 3.4 Hue and saturation in HSI color model (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition,298*)

3.1.2 Color Models Conversions

3.2.2.1 RGB to HSI

HSI model colors are defined based on normalized R G B values (Gonzales & Woods, 2002).

$$r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad b = \frac{B}{R + G + B}$$

Intensity of component;

$$I = \frac{1}{3} (R + G + B)$$

Hue H of any color point P is the angle of color vector with respect to red axis

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{[(R-G)^2+(R-B)(G-B)]^2}} \right\} \text{ Gives angles } 0 \leq H \leq 180 \text{ for } G \geq B$$

If $B > G$ then $H = 360 - H$

$$S = 1 - \frac{3 * \min (R, G, B)}{R + G + B}$$

After these calculations we can find the correct values for HSV color model

3.2.2.2 HSI to RGB

HIS to RGB color space conversion is done with following procedure;

- If $0 < H \leq 120$: $b = \frac{1}{3}(1 - S)$, $r = \frac{1}{3}\left[1 + \frac{S \cdot \cos H}{\cos(60-H)}\right]$, $g = 1 - (r + b)$
- If $120 < H \leq 240$: $H = H - 120$, $r = \frac{1}{3}(1 - S)$, $g = \frac{1}{3}\left[1 + \frac{S \cos H}{\cos(60-H)}\right]$,
 $b = 1 - (r + g)$
- If $240 < H \leq 360$: $H = H - 240$, $g = \frac{1}{3}(1 - S)$, $b = \frac{1}{3}\left[1 + \frac{S \cos H}{\cos(60-H)}\right]$,
 $r = 1 - (b + g)$

To find R G B values $R = 3Ir$, $G = 3Ig$, $B = 3Ib$.

3.1.3 Smoothing Filters

The output of a smoothing is the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters and low pass filters. Filtering a gray level image with smoothing filters results an image with reduced “sharp” transitions. Smoothing filter is mainly used for noise reduction processing and smoothing of false contours that result (Gonzales & Woods, 2002).

1	1	1
1	1	1
1	1	1

Figure 3.5 3×3 Smoothing filter mask

Figure above shows a 3×3 smoothing filter. Using this filter yields standard average of the pixels under the mask.

General implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ (m and n odd) is; (f is image and w is mask)

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

3.2 Some Basic Relationship Between Pixels

3.2.1 Neighbors of a Pixel

A pixel p at coordinates (x, y) has four *horizontal* and *vertical* neighbors whose coordinates are given by $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$. This set of pixels, called the *4-neighbors* of p , is denoted by $N_4(p)$. Each pixel is a unit distance from (x, y) , and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image. The four *diagonal* neighbors of p have coordinates $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ and are denoted by $N_D(p)$. These points, together with the 4-neighbors, are called the *8-neighbors* of p , denoted by $N_8(p)$. As before, some of the points in $N_D(p)$ and $N_8(p)$ fall outside the image if (x, y) is on the border of the image (Gonzales & Woods, 2002).

3.2.2 Connectivity

Connectivity (Gonzales & Woods, 2002). is important concept that is used in establishing boundaries of object and components of area between pixels. To determine connectivity between two pixel it must be determined if they are adjacent if image is binary image ,if image is gray level image then their gray levels must satisfy a specified rules of similarity.

It's considered that there are three type of connectivity

- 4-Connectivity: two pixels A and B those values X are 4-connected if B is the set $N_4(p)$.

- 8-Connectivity : Two pixel A and B those values X are 8-connected if B is the set $N_8(p)$
- M-Connectivity: Two pixel A and B are connected if;
 1. B is in $N_4(p)$.
 2. B is in $N_8(p)$ and the set $N_4(p) \cap N_8(p)$ is empty

3.3 Moment Based Image Analysis

Definition of moments of gray value function $f(x, y)$ of an object is following (Gonzales & Woods, 2002).

$$m_{p,q} = \int \int x^p y^q f(x, y) dx dy$$

Integration is calculated over the area for the object. Every pixel based feature can be used for calculation instead of gray value.

Moments are termed by order of moments. The order of a moment depends on the indices p and q of the moment $m_{p,q}$, $p + q$ is the order of the moment.

- Zero Order Moment

$$m_{0,0} = \int \int f(x, y) dx dy$$

- First Order Moment

$$m_{1,0} = \int \int x f(x, y) dx dy$$

$$m_{0,1} = \int \int y f(x, y) dx dy$$

- Second Order Moments

$$m_{2,0} = \int \int x^2 f(x, y) dx dy$$

$$m_{0,2} = \int \int y^2 f(x, y) dx dy$$

$$m_{1,1} = \int \int xyf(x,y)dxdy$$

These calculations give us spatial moments of the object. Central moments can be found by reducing the spatial moments with center of gravity (x_c, y_c) of the object. Central Moments are center of gravity of the object.

Central moments are calculated as follows:

$$\mu_{p,q} = \int \int (x - x_c)^p (y - y_c)^q f(x,y)dxdy$$

The moments are features of the object, which allow a geometrical reconstruction of the object. They do not have a direct understandable geometrical meaning, but usual geometrical parameters can be derived from them (Gonzales & Woods, 2002).

- Zero order moment $m_{0,0}$ is the area A of the object;

$$A = m_{0,0}$$

- Coordinate x_c and y_c of center of gravity of object can be formulated as first order moments divided by zero order moment

$$x_c = \frac{m_{1,0}}{A} = \frac{m_{1,0}}{m_{0,0}}$$

$$y_c = \frac{m_{0,1}}{A} = \frac{m_{0,1}}{m_{0,0}}$$

- Main Inertial axis could be derived by calculating the eigen values of the inertial tensor (Amit & Mataric, 2002)

$$\lambda_{1,2} = \sqrt{\frac{1}{2}(\mu_{2,0} + \mu_{0,2}) \mp \sqrt{4\mu_{1,1}^2 - (\mu_{2,0} - \mu_{0,2})^2}}$$

Main inertial axes of the object correspond to the semi-major and semi-minor axes a and b of the image ellipse; they can be used as an approximation of the considered object. Main inertial axes are those axes, around which the object can be rotated with

minimal(major semi-axis) or maximal(minor semi-axis) inertia (Gonzales & Woods, 2002).

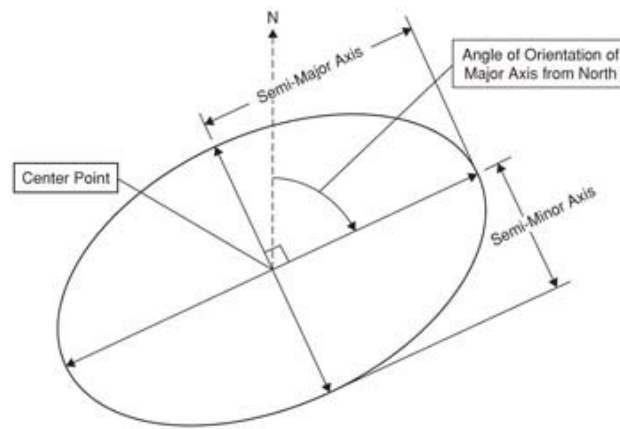


Figure 3.6 Semi axes and orientation

- The orientation θ of the object is defined as the tilt angle between the x-axis and the axis, around which the object can be rotated with minimal inertia (i.e. the direction of the major semi-axis "a"). This corresponds to the eigenvector with minimal eigen value. In this direction the object has its largest extension. It is calculated as follows:

$$\theta = \frac{1}{2} \tan^{-1} \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}$$

Tilt angle θ is the angle between x-axis and semi major axis. Principal value of the arc tangent is chosen such that $-\frac{\pi}{2} \leq \tan^{-1} x \leq \frac{\pi}{2}$

Summarization of orientation with tabulation is given in Table 3.1.

Table 3.1 Summary of orientation calculation

$\mu_{2,0} - \mu_{0,2}$	$\mu_{1,1}$	θ	
Zero	Zero	0^0	
Zero	Positive	$+45^0$	
Zero	Negative	-45^0	

Positive	Zero	0^0	
Negative	Zero	-90^0	

Table 3.1 Continue

Positive	Positive	$\frac{1}{2} \tan^{-1} \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}$	$0^0 < \theta < 45^0$
Positive	Negative	$\frac{1}{2} \tan^{-1} \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}$	$-45^0 < \theta < 0^0$
Negative	Positive	$\frac{1}{2} \tan^{-1} \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} + 90^0$	$45^0 < \theta < 90^0$
Negative	Negative	$\frac{1}{2} \tan^{-1} \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} - 90^0$	$-90^0 < \theta < 45^0$

3.4 Edge Detection

Edge is boundary between two regions with relatively gray level properties. Most edge detection techniques are based on computation of local derivative operation and calculation. Assumption is that regions in the image are sufficiently homogenous so transitions in the regions can be determined in basis on gray levels (Gonzales & Woods, 2002).

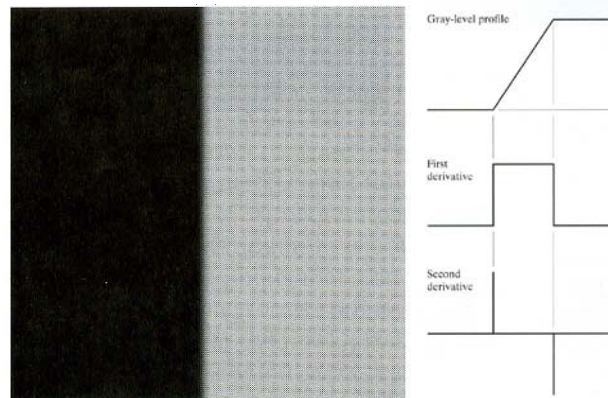


Figure 3.7 Vertical edge separated two region & their edge profile details (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition, 572*)

The first derivative is obtained by using the magnitude of the Gradient at any point. The second derivative is obtained by using the Laplacian.

3.4.1 Gradient Operators

2-D gradient is the base of derivative of a digital image. The gradient of an image $f(x, y)$ at location (x, y) is defined as the *vector* (Gonzales & Woods, 2002).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

It's known that gradient vector points in the direction of maximum rate of change of f at coordinates (x, y) . Magnitude of this vector is important quantity in edge detection;

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}$$

This equation gives us the maximum rate of increase of $f(x, y)$ in the direction of ∇f at (x, y) . ∇f Is *gradient*. Let $\alpha(x, y)$ is the direction angle of gradient at (x, y) .

$$\alpha(x, y) = \tan^{-1} \left[\frac{G_x}{G_y} \right]$$

Gradient of an image is based on partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location. angle is measured with respect to the x-axis. Direction of an *edge* is perpendicular to the direction of gradient vector at that point (Gonzales & Woods, 2002).

-1	0	0	-1	-1	-1	-1	-1	0	1	-1	-2	-1	-1	0	1
0	1	0	0	0	0	0	-1	0	1	0	0	0	-2	0	2
0	1	1	0	1	1	1	-1	0	1	1	2	1	-1	0	1

Figure 3.8 Robert, Prewitt and Sobel Masks for first-order partial derivatives (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*,134-136)

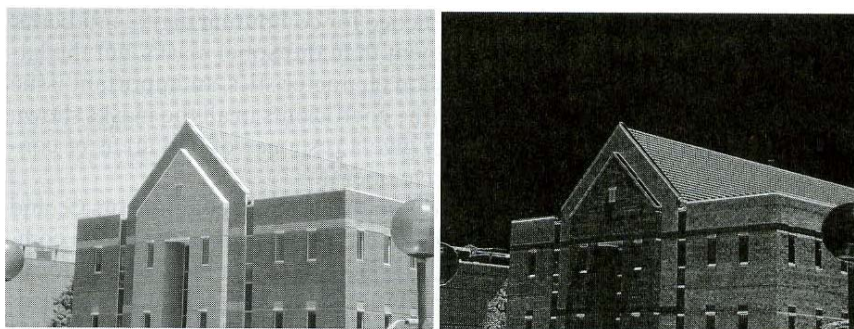


Figure 3.9 Original image and $|G_x|$ component of gradient in x-direction (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 134-136)

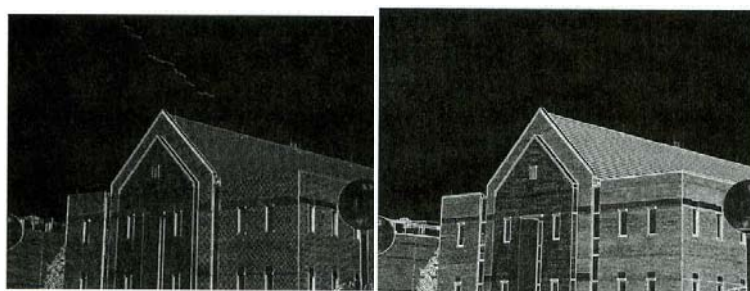


Figure 3.10 $|G_y|$ component of gradient in y-direction & $|G_y| + |G_y|$ (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 134-136)

3.4.2 Laplacian Operators

Laplacian is second order derivative of 2-D $f(x, y)$ function (Gonzales & Woods, 2002).

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Mask for calculating Laplacian is shown at below;

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Figure 3.11 Laplacian masks (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 128-134)

There are some disadvantages of Laplacian;

- Unacceptably sensitive to noise
- Produces double edges when the magnitude of the Laplacian is thresholded to detect edges
- Unable to detect edge direction.

3.5 Zooming-Shrinking Digital Images

Zooming and shrinking are applied digital images. Zooming may be viewed as oversampling and shrinking may be viewed as under sampling (Gonzales & Woods, 2002).

3.5.1 Zooming

Zooming has two steps these are creation of new pixel locations and assignment the pixel values to these locations. There are two kind of zooming algorithm (Gonzales & Woods, 2002).

3.5.1.1 Nearest Neighbor Interpolation

Suppose that we have an image of size 500×500 and we want to enlarge it 1.5 times to 750×750 pixels. In NNI algorithm the concept is simple lying 750×750 grid over the original image. We can sense that spacing in the grid will be less than one pixel, in order to assign new pixel values on the overlay, nearest pixels to the grid value is taken.

3.5.1.2 Pixel Replication

This algorithm is used only for if we want enlarge the image for integer number of times. Algorithm can be explained by example. Suppose that Image is need to enlarge double size, we can duplicate each column. This doubles the image size in the horizontal direction. Then, we duplicate each row of the enlarged image to double the size in the vertical direction

3.5.2 Shrinking

Digital Image Shrinking is done in a similar way of Zooming. For the NNI we expand the grid to fit the original image and takes nearest pixels value, to prevent the aliasing its good blurring image before this algorithm, the other algorithm that is similar to Pixel Replication is deleting the rows and columns instead of replicating them.

3.6 Image Rotating

This algorithm is “Mapping” the position (x_1, y_1) of a picture element in an input image onto a position (x_2, y_2) in an output image by rotating it through a user-specified angle θ about an origin O . Rotation is most commonly used to improve the visual appearance of an image, although it can be useful as a preprocessor in applications where directional operators are involved.

$$x_2 = \cos(\theta) \times (x_1 - x_0) - \sin(\theta) \times (y_1 - y_0) + x_0$$

$$y_2 = \sin(\theta) \times (x_1 - x_0) + \cos(\theta) \times (y_1 - y_0) + y_0$$

(x_0, y_0) Are the coordinates of the center of rotation and θ is the angle of rotation with clockwise rotations having positive angles. Even more than the translation operator, the rotation operation produces output locations (x_2, y_2) which do not fit within the boundaries of the image. In such cases, destination elements which have been mapped outside the image are ignored by most implementations. Pixel locations out of which an image has been rotated are usually filled in with black pixels.

3.7 Morphological Operation-Region Filling

Next we develop a simple algorithm for region filling based on set dilations, complementation, and intersections... A denotes a set containing a subset whose elements are 8-connected boundary points of a region. Beginning with a point p inside the boundary, the objective is to fill the entire region with 1's:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3 \dots$$

Where $X_0 = p$ and B is the symmetric structuring element...The algorithm terminates at iteration step k if $X_k = X_{k-1}$, the set union of X_k and contains the filled set and its boundary. (Rafael C. Gonzalez, 2001 p.535)

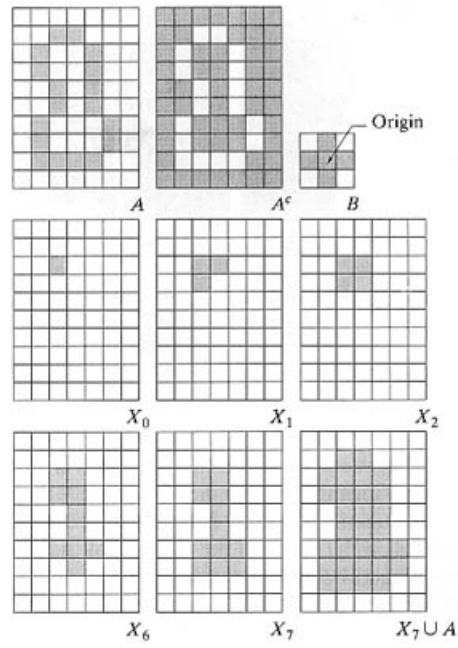


Figure 3.12 Region Filling (Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 535-536)

CHAPTER FOUR

ROBOTIC ARM

4.1 Properties of Robotic Arm

Robotic arm used for this thesis is shown below;

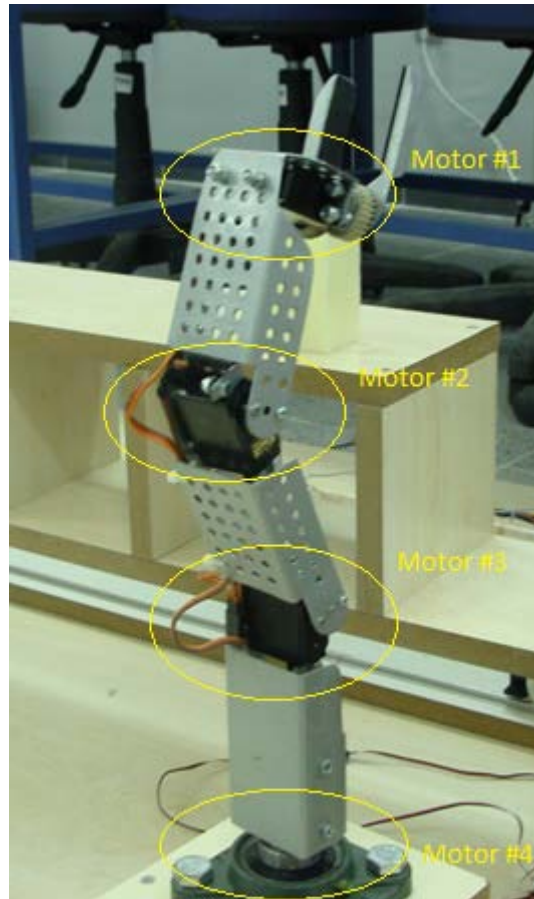


Figure 4.1 Picture of robotic Arm

As shown in the picture the robotic arm contains four motor as three for joints and one for gripper. Motor #4 is placed in the arm to have rotating property maximum 180°

4.2 Properties of Parallax Servo Controller Card

Robotic arm is controlled by this card that contains 16 channels output for servos and ability of communicating with another parallax to control up to 32 servo motors using just one serial communication. Parallax has its own communication protocol by using this protocol besides the rotating any motor, user can define the even turning speed of the motor.

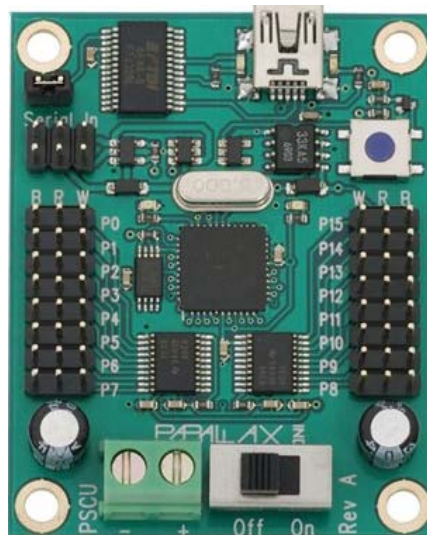


Figure 4.2 Parallax servo controller card

CHAPTER FIVE

SEGMENTATION OF HAND

5.1 Program Algorithm

Full Algorithm of the program is given below; Detailed information about Calibration Algorithm and Gesture Recognition Algorithms can be found under related headers.

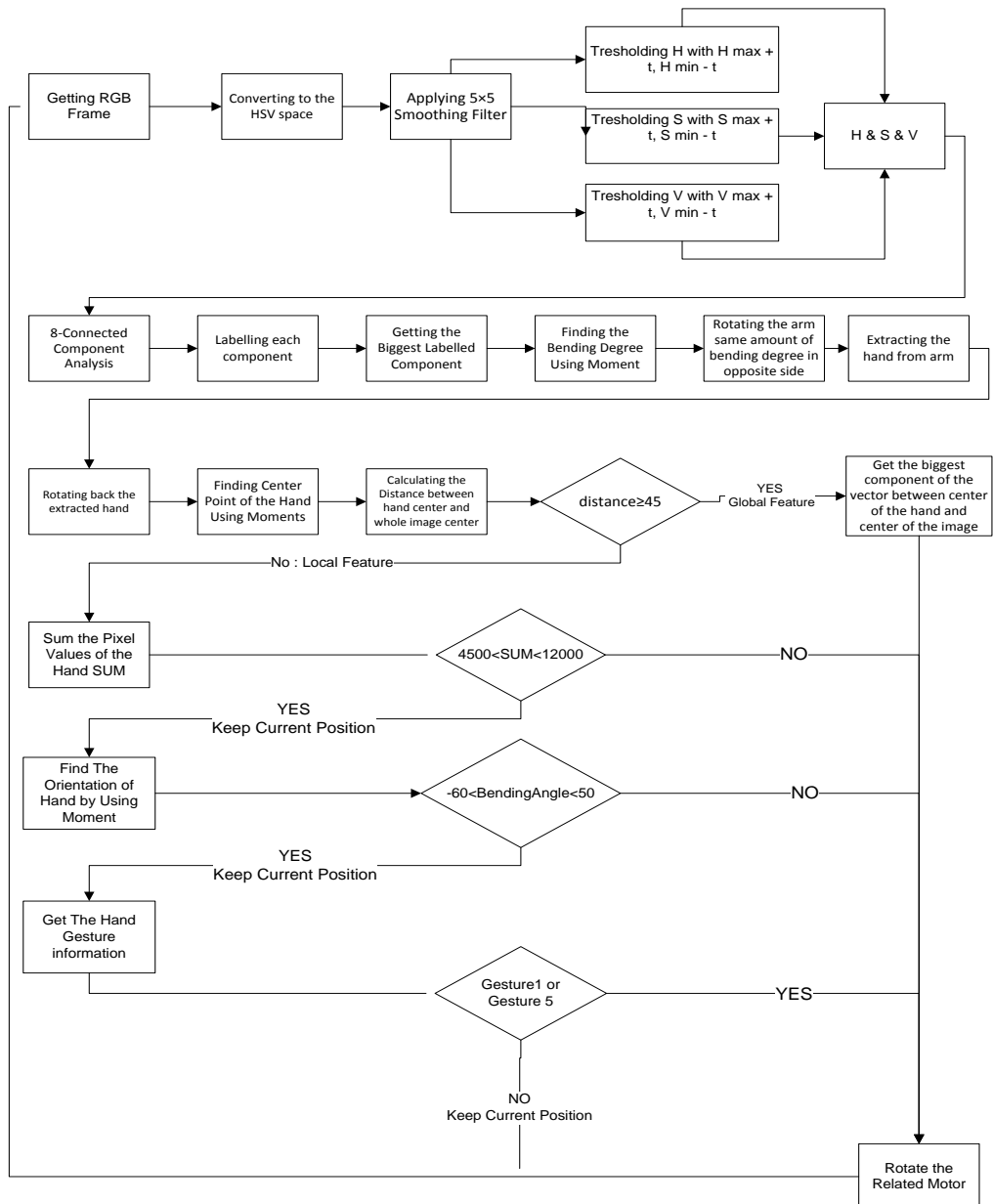


Figure 5.1 Algorithm of the program.

5.2 Calibration

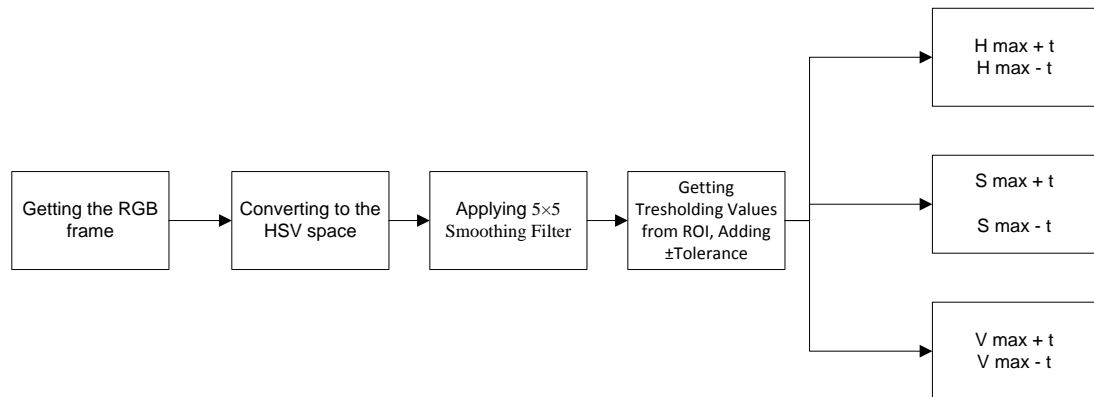


Figure 5.2 Calibration algorithm

Calibration before using is a very important step for this research. The aim of the calibration is getting stable calculations and results against changing light, user or working area conditions.

The process is as follows: The user puts his/her hands on region of interest of the screen then waits for the shoot. After getting photo of the hand, first RGB values of area of hand are taken then it is converted to HSI values. As mentioned before, HSI color space gives better results for skin detection.



Figure 5.3 Getting calibration shoot of hand

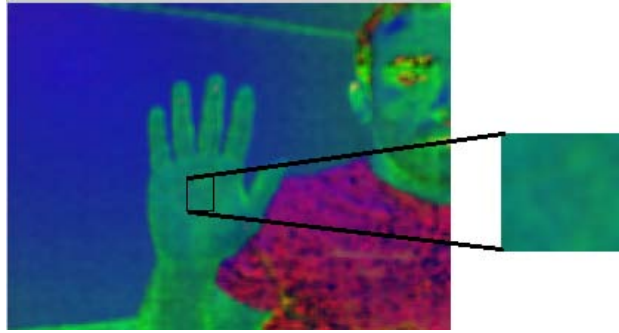


Figure 5.4 Converting the Calibration picture to HSV and taking the Threshold values.

After low pass filtering and taking HSI hand values, they are for thresholding. These values are stored on computer with some tolerance which is ± 0.07 . After that step, minimum and maximum HSI values of the stored data can be taken for thresholding. Therefore we have values for:

$$H_{max} \ H_{min}, \ S_{max} \ S_{min}, \ I_{max} \ I_{min}.$$

5.3 Video Processing

Video part of the research is explained in detail step-by-step below, algorithm flow chart can be found and of the chapter.

5.3.1 Getting Frame and Tresholding

Major steps in the visual processing stage can be summarized as below until tresholding

1. Getting frame in RGB color space and converting it to HSV color space for better color differentiation for skin and other elements of frame.

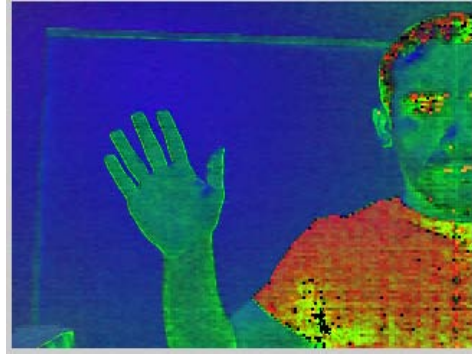


Figure 5.5 HSI color space converted image

2. Using averaging filter to smooth the frame and remove environmental and camera noise 5×5 filter is used.



Figure 5.6 Filtered HSI colors

3. Thresholding is applied to H, S and I segment of the image separately with some tolerance $\mp t$ all values that are not in thresholding range are set to zero. After thresholding a new full black binary same size image is created and for each H, S and I. Thresholded pixel coordinates maps the new binary image pixels. For example let H_t be created binary image for H channel of the frame. Then

$$H_t[i, j] = \begin{cases} 1, & \text{if } H_{min} - \mp t < H[i, j] < H_{max} + \mp t \\ 0, & \text{otherwise} \end{cases}$$



Figure 5.7 *I* color array of image and tresholded&mapped binary image

4. After each H, S and I are mapped to binary images separately they are processed to get rid of noise...



Figure 5.8 *H & S & I* binary Image.

5.3.2 Determination Location of Hand

1. After getting tresholded binary image from the frame, it is analyzed for 8-connecticity. After analyzing all the connected components of binary image, they are compared with each other based on the amount of pixel on each component.



Figure 5.9 8-Labelled and colored connected components

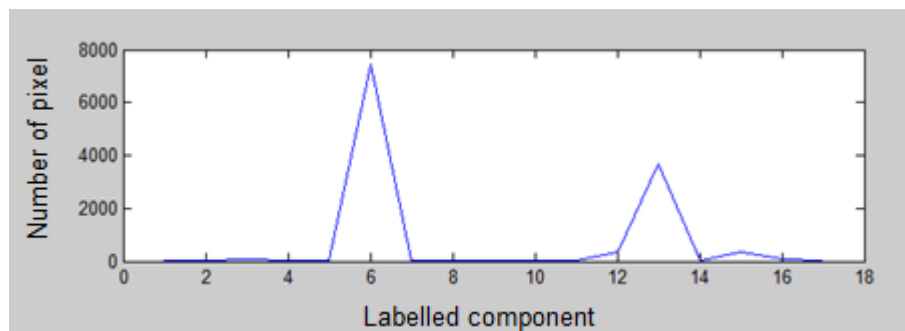


Figure 5.10 Graph versus label name and amount of pixel of that label

- It's concluded that because of position of hand and its distance from camera, Label of hand always gives the maximum number of pixels in the graph above unless hand is too far beyond. All labels except hand is cleared from the binary image based on that conclusion.

5.3.3 Elimination of Arm

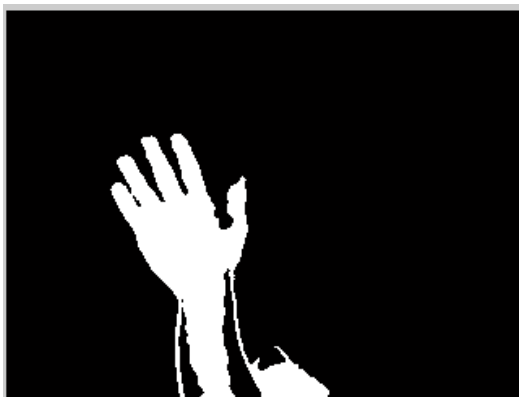


Figure 5.11 Cleared binary frame after getting number of pixel analysis

One of the most important part of the study is given in this part. Algorithm that is used for separating arm from hand needs orientation of hand after finding bending angle of arm. Then Image is rotated to opposite site to have a straight arm image. If the angle is for example -65° to the right, image is rotated $+65^{\circ}$ to the left. This causes the hand position to be perpendicular. After getting horizontal position of hand, the analysis is started. Every two rows at a 25 pixels distance from each other are summed from down to up (from arm to hand) and their ratio is checked the reason of choosing 25 is that there is always guarantee case which is when is lower row is at the arm other one is always at the wrist. For example S_a is the summation of first row, and then S_{a-25} is the next one. When $S_{a-25}/S_a \geq 1.6$ then all the rows after S_{a-25} are cleared. This algorithm gives good performance for getting rid of the arm image. Then the rotated binary image is rotated to first position; for example, to the -65 degree back. Finally the image is cropped to $320*240$ size.

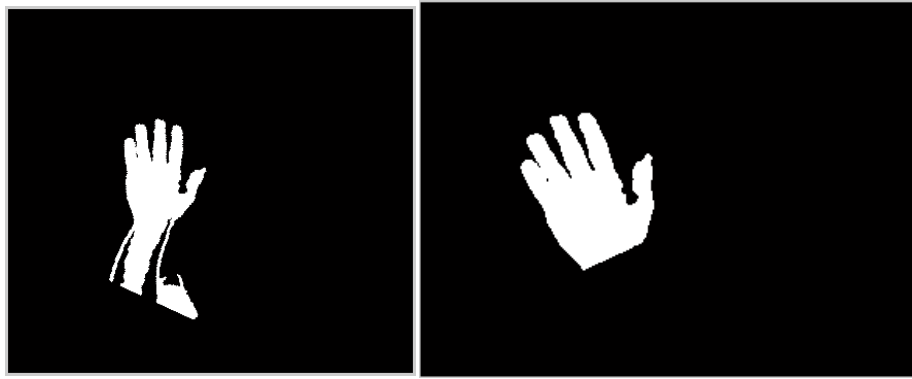


Figure 5.12 Rotated image for deleting-arm algorithm.

As mentioned before, the system has both Local and Global features. Those features are calculated according to position of the hand. If the distance between the centroid of hand and the centroid of binary image is smaller than 45 unit distance then Local Features are calculated which contains “Gesture Analyzing”, “Bending Degree”, “Distance of Hand from Camera”. Otherwise, Global Features are used which calculates “Position of hand” (not distance) on the binary image.

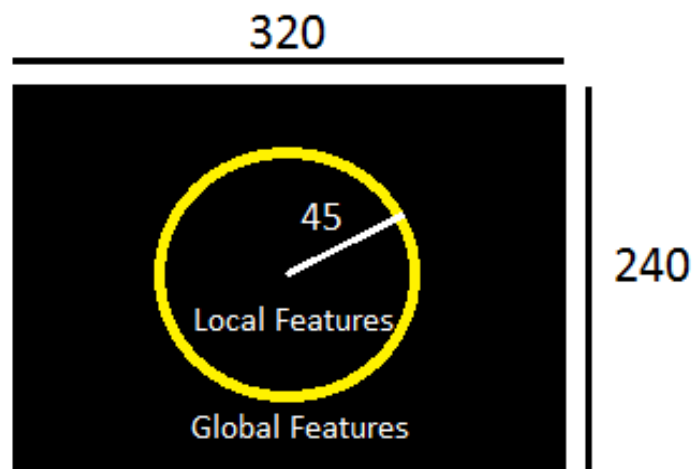


Figure 5.13 Extraction area of each feature group

CHAPTER SIX

HAND GESTURE RECOGNITION

Because Gesture Detection is a very important part of this project, it would be better to explain both approximations of gesture detection under distinct headers instead of explaining under Local Feature Extraction headers later. Gesture Recognition algorithm is given below;

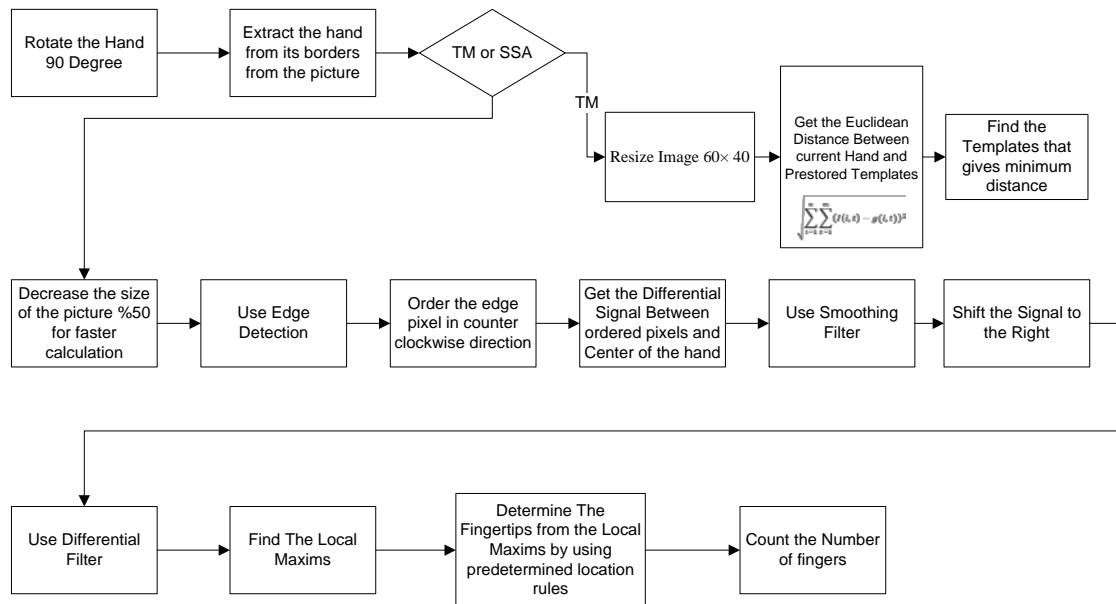


Figure 6.1 Gesture recognition algorithm of both methods

6.1 Template Matching Algorithm

Template Matching Algorithm (TMA) basically consists of transforming the hand into canonical frame and comparing the image data with prestored data that had been prepared before.

In mathematics, a canonical form (often called normal form or standard form) of an object is a standard way of presenting that object.

In computer science, canonicalization (also sometimes standardization or normalization) is a process for converting data that has more than one possible representation into a "standard", "normal", or canonical form.

Using hand, yaw and scaling information, it's possible to have an image memory which consists of hand gesture frames in the same yaw angle and scaling (this is called canonical frame) (Wikipedia, 2011).

6.1.1 Evaluation of Canonical Frame

First property of all canonical frames is that they are perpendicular with a yaw degree (0^0), which is basically rotating image same amount of yaw angle to the opposite side. The second property of them is that they have constant size of 60×40 image. Before resizing the image; an image filling algorithm is used to avoid any "holes" in the picture. This would be an undesired situation.



Figure 6.2 Processing steps of canonical frame these are segmentation of hand image filling, rotating and resizing

Canonical templates are prepared for the TMA which is shown below. Template Base consists 17 canonical frames there are five sign all these are 1, 2,3,4,5.

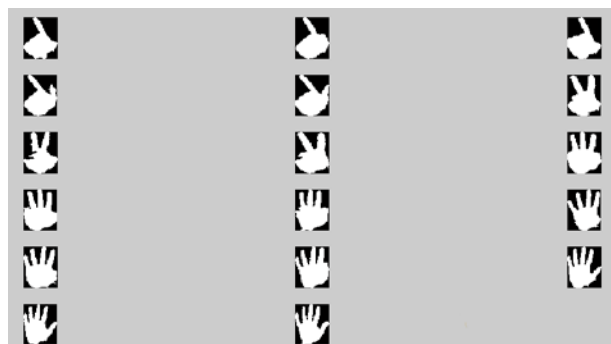


Figure 6.3 Canonical template base.

For robotic arm gripper, Gestures 1 and 5 will be used; 1 is to close the gripper, 5 is to open the gripper

6.1.2 Applying the Template Matching Algorithm

After having a canonical frame and Template base, Euclidean distance between frame and all frames in the base are calculated. The gesture which gives the minimum distance is the Recognized Gesture. Euclidean Distance can be found by using the formula below:

$$\sqrt{\sum_{i=1}^n \sum_{t=1}^m (I(i, t) - g(i, t))^2}$$

Applying Euclidean Distance to the base gives us the distance information below

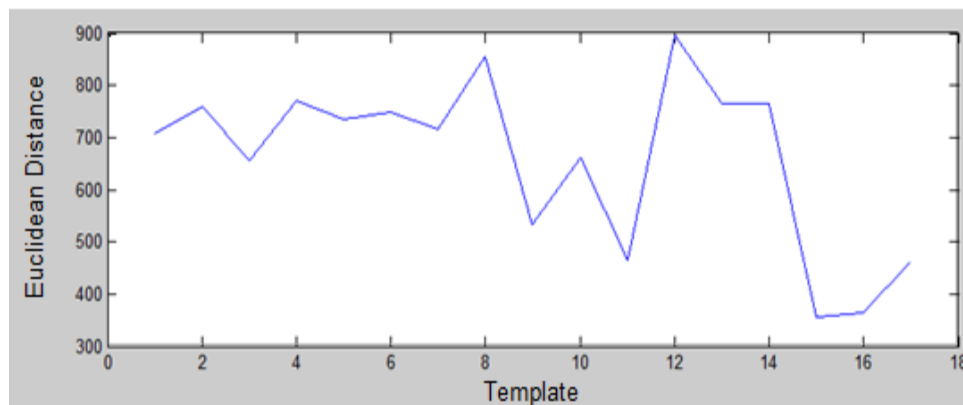


Figure 6.4 Distance information of templates

As it's shown in the graph, minimum distance is given in 15th template which is gesture five

6.2 Signature Signal Algorithm

Signature Signal Algorithm (SSA) is a 1-D functional representation of boundary by taking the distance between each boundary pixel and centroid of the hand in clockwise direction. Generated signatures are dependent on "Rotation" and "Scaling" of the frame. SSA algorithm used canonical frame algorithm to get rid of rotation dependency until rescaling of hand.

6.2.1 Getting rid of Dependency of Scaling

Dependency of Scaling can be explained simply as length of Signature Signal is depend of the hands distance between camera, so when hands get close the camera length of the signal increases. There is one more problem which is caused from the matlab too, as experimented from the program matlab automatically stops calculating while online processing if the signature signal length is bigger than 120. To get rid of these problems there are some extra process after rotating of hand. Second problem handled first, after rotating the hand ,it always rescales by %50 smaller size no matter actual size of hand, as experiment while programming this solution helps us maximum 80 or 90 length of signal ,this amount of length does not produce stopping on program. Having standardization on length of Signature Signal helps the researcher to have more research ways, for example to use neural network or comparing signals with each other in some ways (such as power spectral density) always needs same length of signal.



Figure 6.5 All preprocessing steps including resizing %50

After resizing image %50 smaller, a new image extracted from that image, which touches border of hands, this process causes less calculation time. Image is shown below

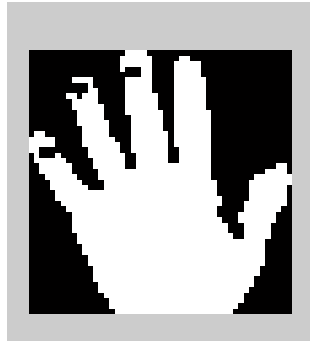


Figure 6.6 Last situation of hand before Edge Detection

6.2.2 Signature Signal Extraction

Before having signature signal there is need of ordering the edge pixel address as counter clockwise direction.



Figure 6.7 Edge detected hand image

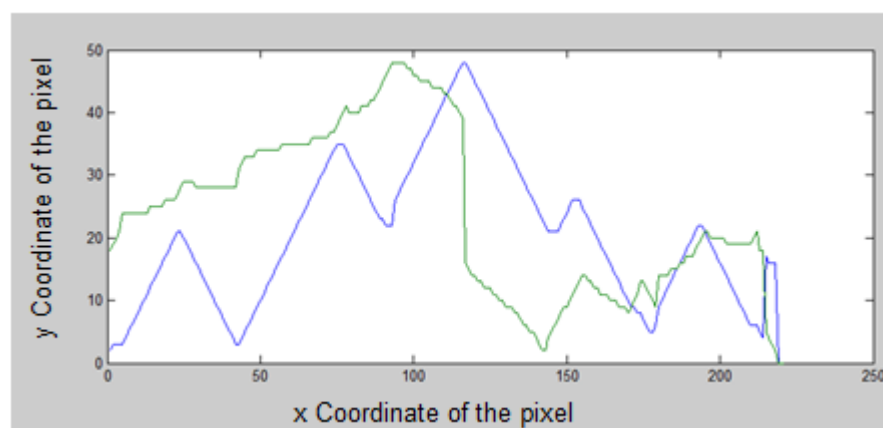


Figure 6.8 Ordered edge pixel address in counter clockwise direction

Ordering of pixels is started from the middle of fingertip of middle fingers to the again middle finger, in clock wise direction. As mentioned before calculation this much of length takes too much time and may cause the matlab to stop, to overcome this problem signal is decimated by rate of 2.

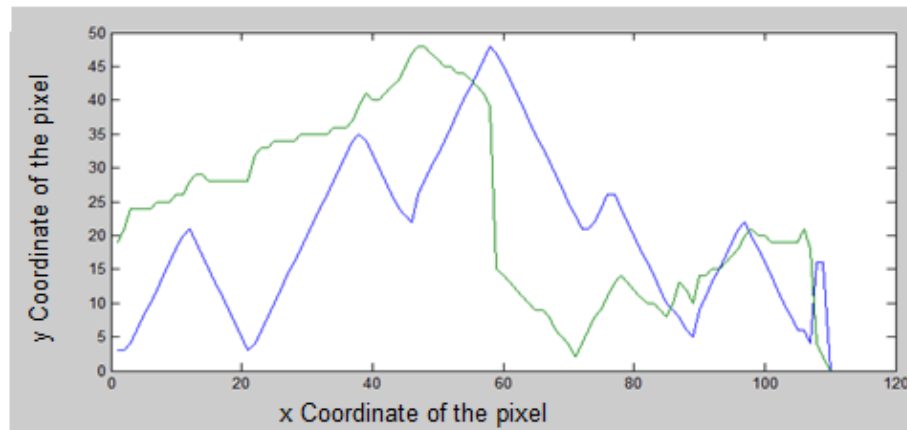


Figure 6.9 Decimated pixel address

After decimation of the ordered address signal a new signal is extracted that is the distance from ordered address and center of hand by using distance formula of two points. (Note: For better performance center of hand is assumed as lower of real centroid of rate of $3/4$).

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

This technique gives better looking for peaks which is the place of fingertips. After having distance signal, signal's length is changed to 80 for constant length. Decimation or interpolation are used depend on length of signal, and then filtered for noises. Decimation or interpolation are used depend on length of signal, and then filtered for noises.

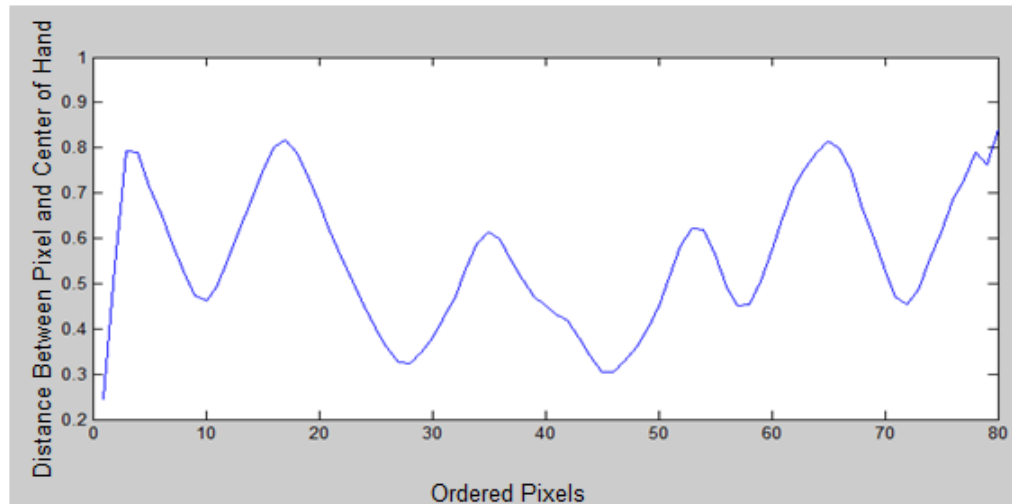


Figure 6.10 Distance between ordered addresses and centroid of hand.

After shifting the previous signal, Signature Signal is created. As mentioned in sec 6.2.1 there is a scaling dependency problem of our signal, Normally that could be overcome while scaling %50 factor time, it could be a solution scaling a constant scale (for example 60×40) instead of scaling %50. Because of the controlling of gripper depends of Gesture of 1 and Gesture of 5, its considered to be better counting of fingers for hand gesture detection.

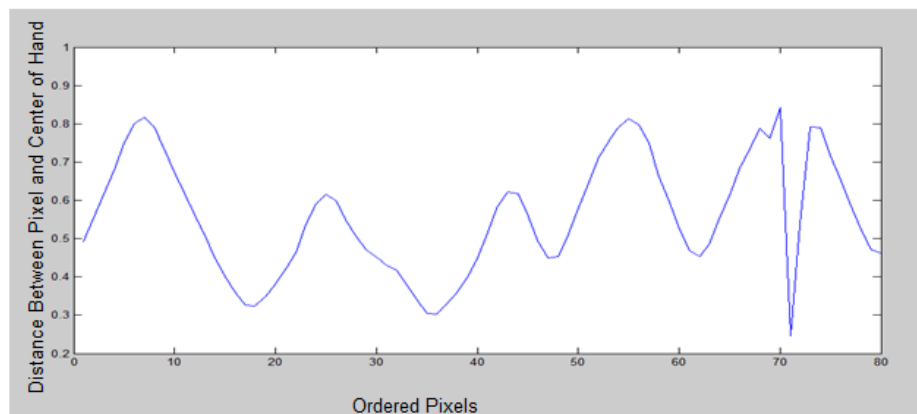


Figure 6.11 Shifted signature signal

Place of fingers must be identified to count them. Fingertips are placed as peaks (local maxims) on Signature signal, to find peaks, differential must be found by the formula below

$$f(x) = y \rightarrow f'(x) = \Delta y / \Delta x$$

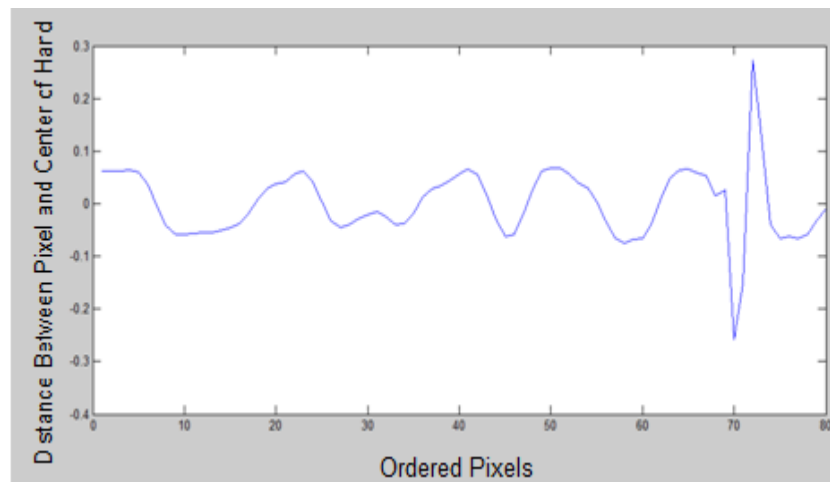


Figure 6.12 Gradient of signature signal.

After getting gradient, Local Maxims can be identified as Positive-0-Negative directions

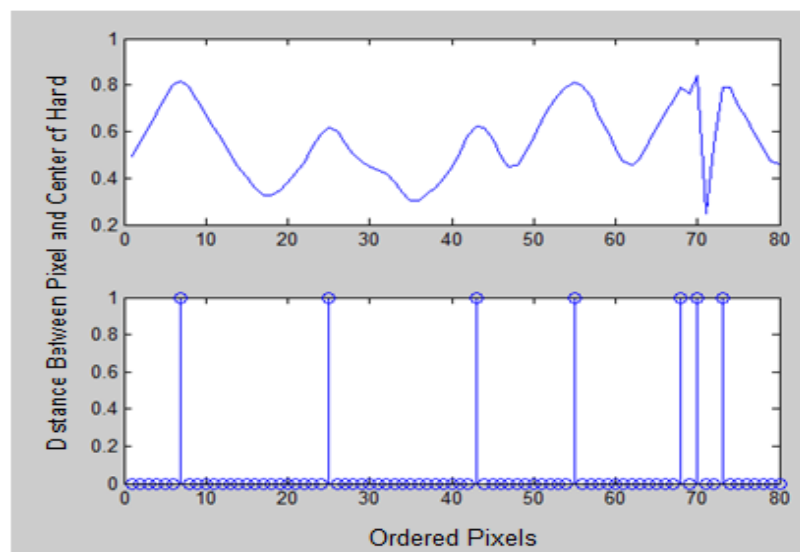


Figure 6.13 Local maxims

As it seems there are 7 Local maxims instead of finding 5 ,That problem is solved by programming using these criterion, Minimum Peak Height =0.5, Minimum Peak Distance=5, Next graph shows the situation of local maxims, after using these criterion;

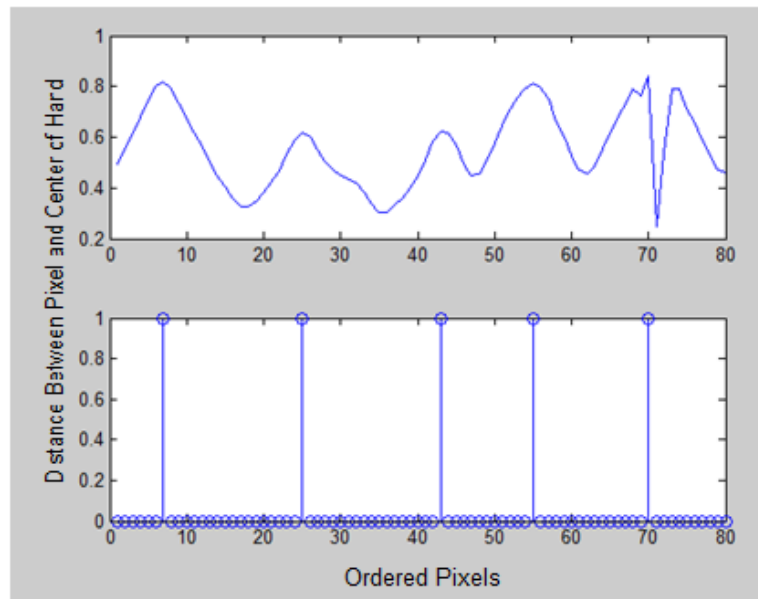


Figure 6.14 Shifted signal it can be seen that there is a interrupted, broken peak ,this peak caused of middle finger, actually that problem can be overcome by using some algorithm while writing the program after derivative.

So scaling dependency is got rid of while creating a normalized signal that shows the place of fingers from the previous signal, which is shown below;

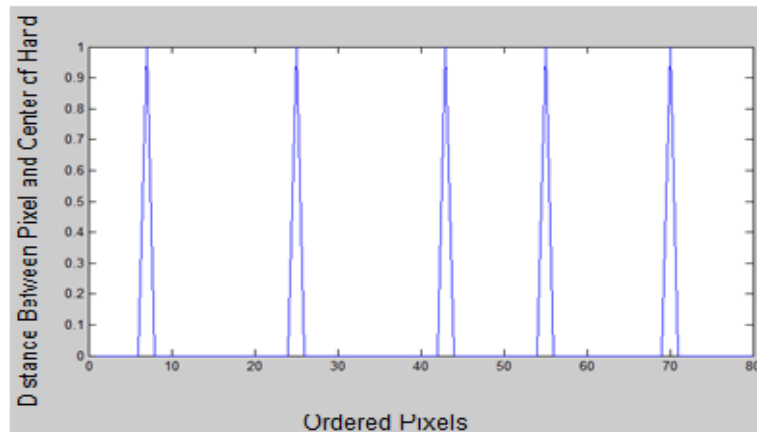


Figure 6.15 Finger place signals current gesture is 5

As it's shown in the Fig 6-9 gesture can be recognized by summing the peaks.

6.3 Performance Analysis of Algorithms

Both algorithm was tested in same conditions, Light source was in the left side and the corner on the ceil according to the hand. This kind of source position always causes to be shadow in the observed object. It was a good solution to use greater tolerance $\bar{\tau}t$ to get rid of the shadows.

6.3.1 Analysis for Template Matching Algorithm

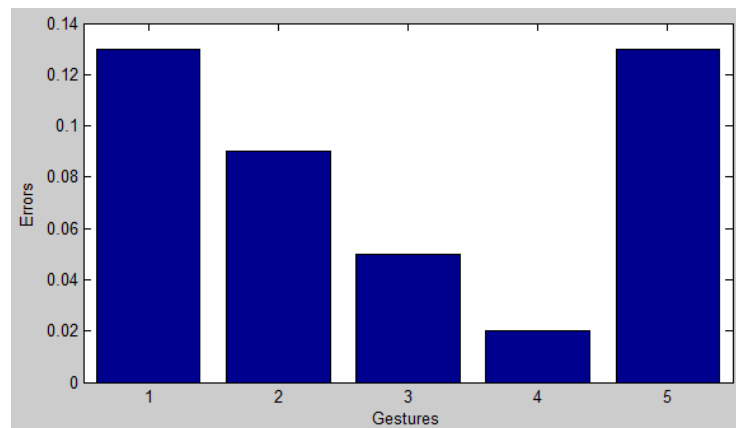


Figure 6.16 All gestures performance analysis for TMA

6.3.2 Analysis for Signature Signal Algorithm

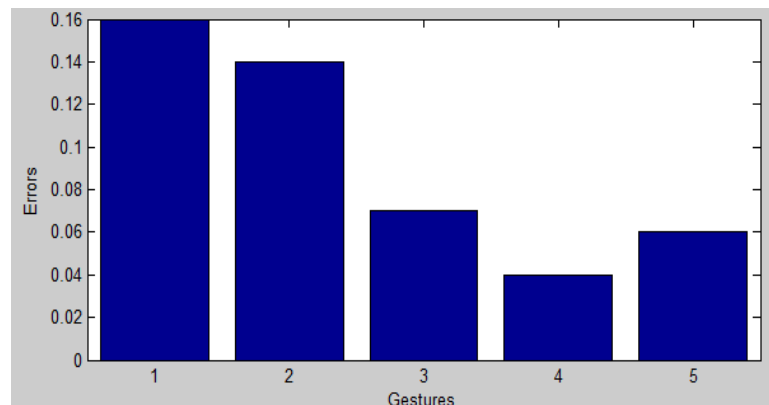


Figure 6.17 All gestures performance analysis for SSA

CHAPTER SEVEN

FEATURE EXTRACTION

Controlling of a robotic arm by Gesture Recognition needs well prepared Feature Extraction. There are 10 control bits for the robotic arm first six bits are taken from local features, rest of them are taken from global features.

7.1 Local Feature Extraction

Local features are extracted if segmented hand is at a distance of maximum 45 units from the center of whole image shown before. Local features produces first 6 bits (b1, b2, b3, b4, b5, b6) of control bits which will be used for the control signal for three joints of robotic arm. Control bits have priority, which mean there are just one move in a time so motionless motors keep their last position while active motors takes new position. Obtainment of first six control bits driven from Local features are explained step-by-step below;

- First priority is “Summing Pixel Values”(SPV) which is used for measuring distance of hand to the camera by using the;

$$\sum_{i=1}^n \sum_{t=1}^m I(i, t)$$

Distance Distance of hand found by SPV controls third and fourth bits of control bits (b3 and b4);

$$(b3, b4) = \begin{cases} (1,0) & \text{if Summation} > 12000 \\ (0,1) & \text{if Summation} < 4500 \\ (0,0) & \text{else where} \end{cases}$$

Motor#2 shown at Fig. 4-1 moves depends on situation of these bits

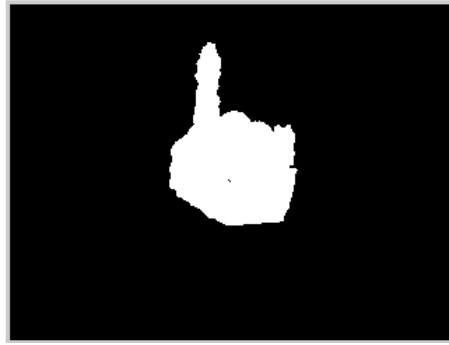


Figure 7.1 Summation of pixel values is 6686

- Second priority is Orientation of hand, Orientation Analysis controls fifth and sixth bits (b5, b6) of control bits. Motor that is controlled by these bits are shown below as yellow joint. Because of Robotic arm doesn't have this joint, the place of it is shown in another robotic arm figure.

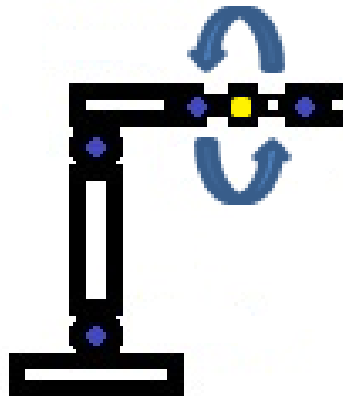


Figure 7.2 Controlled motor and its motion type by orientation



Figure 7.3 Example for orientation, bending angle is -65 degree to the right

Orientation bits control the gripper to turn around itself. These bits can be controlled if summation of pixel values is between 12000 and 4500.

$$(b5, b6) = \begin{cases} (1,0) & \text{if } -60 < \text{Bending Angle} < 0 \\ (0,1) & \text{if } 0 < \text{Bending Angle} < 50 \\ (0,0) & \text{else where} \end{cases}$$

- Third priority is Hand Gesture Recognition which is explained detail in previously. Hand Gesture Recognition controls first and second bits (b1 b2) of control bits. Two different hand gestures are selected for controlling: Gesture5 and Gesture1, Gesture5 is used for opening the gripper, and Gesture1 is used to close the gripper.

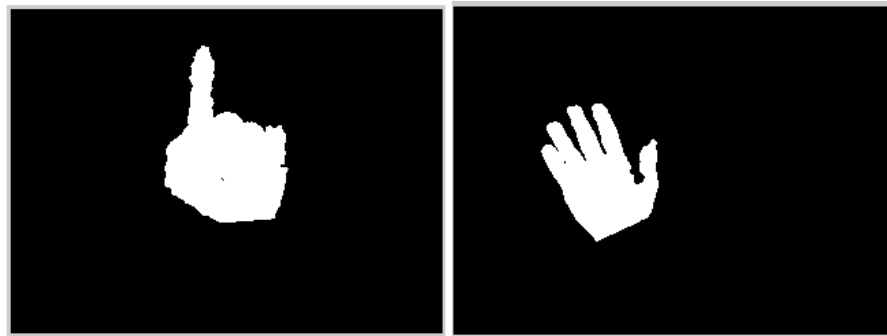


Figure 7.2 Example for Gesture1 and Gesture5

$$(b1, b2) = \begin{cases} (1,0) & \text{if } gesture = \text{Gesture1} \\ (0,1) & \text{if } gesture = \text{Gesture5} \\ \text{previous output} & \text{if } gesture = \text{Another Gesture} \end{cases}$$

Motor#1 shown at Fig. 4-1 moves depend on these bits. These bits can be controlled if summation of pixel values is between 12000 and 4500 and if hand doesn't exceed necessary bending angle.

7.2 Global Feature Extraction

Global Features are extracted if the distance from center of hand and image is bigger than 45 units as shown in Fig.5-12. G.F is used to control Motor#3 and Motor#4 shown at Fig. 4-1. There are 4 directions which are controlled by this feature based on the vector is between the center of hand from the center of whole image.

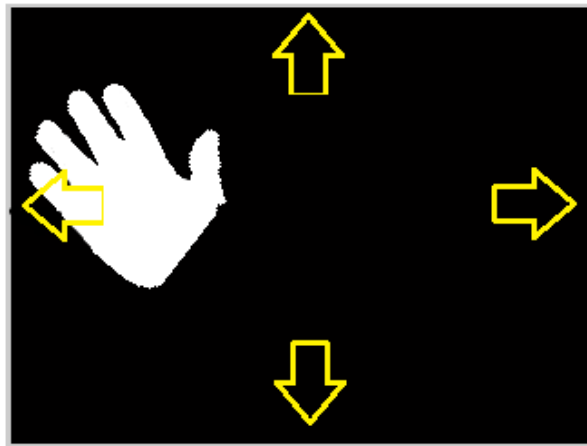


Figure 7.3 Directions chosen by Global Feature.

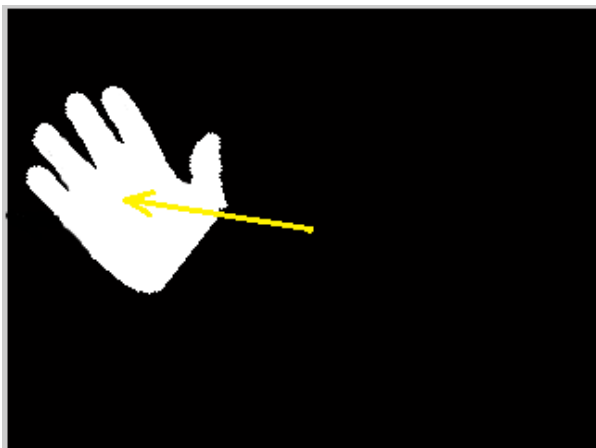


Figure 7.4 Vector of Center of Hand. Scalar components of vector is 69 and 23

After getting scalar components of the vector by using;

$$V_x = V * \cos \phi$$

$$V_y = V * \sin \phi$$

Biggest components plane is chosen based on its own positive or negative axis depending on its magnitude. For example, if biggest components was +70 over x plane, then chosen direction would be towards right.

CHAPTER EIGHT

CONCLUSIONS

8.1 Conclusion

Computer vision based robot controlling is an active research area that covers many different fields of science and engineering such as computer vision, kinematics, and image processing, artificial intelligent. However when controller is human, so many difficulties arise for determining the human dynamics. In order to overcome these difficulties and accomplish robot controlling by hand gesture two different methods have been prepared for gesture recognition. Both methods are applied if the hand is in Local Feature Extraction area and orientation is suitable range, because these methods are used for controlling gripper part of robotic arm.

First method is Template Matching Algorithm, second algorithm is Signature Signal Algorithm, Template Matching Algorithm compares the current hand gesture with pre-stored gestures in memory one-by-one then decided gesture is given which gives minimum Euclidian distance.

Template Matching Algorithm has faster calculation than Signature Signal Algorithm but this fast calculation has some disadvantages which is the requirement for more image memory, this disadvantages brings some recognition problems, it can be seen that computer sometimes return the user Gesture2 instead of Gesture1 when we look detailed analyze results computer sometimes recognize the joint of the thumb as a thumb, if there was more image memory based on templates this recognition error would decrease, requirement for more image memory can also be seen in Gesture5 detailed test results, computer sometimes gives Gesture4 instead of Gesture5 this is because of Gesture4 has more templates than Gesture5 in image memory.

Second method is Signature Signal Algorithm; Signature Signal Algorithm counts fingertips from Signature Signal that is distance signal between the edge pixels and center of hand. There are some digital signal processing steps for detection of fingertips in the signal.

Signature Signal Algorithm is based on the morphological analysis, this kind of approach is susceptible to conditions of the control room, for example if there is not enough light edge detection results may have so many noise which will be counted as fingers, or if background color is very close to skin colors this condition may produce same error, that property brings out that SSA algorithm usually has more error than TMA algorithm, it can be seen that computer returns Gesture2 instead of Gesture1 by looking the detailed test results of SSA algorithm for Gesture 1, this condition is because of non eliminated noise on the edge of the hand which is counted as fingers, this error also can be seen in Gesture2 test results, most of the errors is caused by the unwanted noise on the edge of the hand which leads to get result as Gesture3.

Successfully object recognition is based on firstly successfully object extraction, then rest of the work is based on programmers creativity, power of these algorithms is based on this subject, this thesis used an original approach to real time hand extraction without using any tool, after hand extraction it can be seen that there is a simple image of hand without arm (or any other object) which is ready for analyze of the programmer.

Edge detection Signature Signal Algorithm had been decided to send to Back Propagation Neural Network for the recognition but it was given up after seen that, there were need only for two epochs for the training of the system. This condition shows that it is easy to find the finger tips without any intelligent system.

The reason of choosing these two methods is both methods have complementary properties for each other. Template Matching Algorithm is faster because of having less processing but uses more memory space because of containing visual memory base. Signature Signal Algorithm is slower because of having more calculation and uses less memory steps because of working on memoryless system.

8.2 Proposed Feature Work

Hand segmentation algorithms may be improved to be more robust because hand segmentation produces unwanted solution when hand touches the face, hand and face

are combined and becomes one connected component object after touching, and this problem produces so many difficulties for gesture detection.

Based on experience that is gained through this research, a hybrid mimicking system may be constructed using the advantageous properties of each proposed system while reducing their disadvantages.

Full performance of the program can be tested by using robotic arm that can rotate wrist this situation would produce better performance for controlling.

REFERENCES

- Amit, R., & Mataric, M. (2002). Learning Movement Sequences from Demonstration, *International Conference on Development and Learning*, 203-208
- Bernardo, E., & Goncalves, L., & Perona, P. (1996). Monocular Tracking of the Human Arm in 3D: Real-Time Implementation and Experiments, *Proceedings of the International Conference on Pattern Recognition*, 622-626
- Chang, Y. , & Chen, S. , & Huang, J. (2011). A pilot study for young adults with motor disabilities, *Research in Developmental Disabilities*, 32, 2566-2570
- Cheng, G., & Kuniyoshi, Y. (2000). Real-Time Mimicking of Human Body Motion by a Humanoid Robot, *International Conference on Intelligent Autonomous Systems*, 273-280
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 282-283
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 289-298
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 299-302
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 119
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 66
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 66-70
- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 52-66

- Gonzales, R.C., & Woods, R.E. (2002). *Digital Image Processing By Gonzalez, 2nd Edition*, 535-536
- Littmann, E., & Drees, A., & Ritter, H. (1996). Robot Guidance by Human Pointing Gestures, Proceedings of NICROSP, *IEEE Computer Society Press*
- Microsoft, (2011). Kinect for XBox 360 - Xbox.com, retrieved 21 june 2011, from <http://www.xbox.com/en-GB/kinect>
- Pavlovic, V.I., & Sharma, R., & Huang, T.S. (1997). Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 677- 695
- Rizzo, A., & Lange, B., & Suma, A. E.,& Bolas, M. (2011). New Tools to Address Obesity and Diabetes, *Journal of Diabetes Science and Technology*, 5, 258-264
- Rybski, P.E., & Voyles, R.M. (1999). Interactive Task Training of a Mobile Robot through Human Gesture Recognition, *IEEE International Conference on Robotics and Automation*, 664-669
- Triesch, J., & Von der Malsburg, C. (1998). A Gesture Interface for Human-Robot-Interaction, *IEEE Third International Conference on Automatic Face and Gesture Recognition*, 1998
- Ude, A., & Atkeson, C.G. (2001). Real-Time Visual System for Interaction with a Humanoid Robot, *International Conference on Intelligent Robots and Systems*
- White, K. B., & Searleman, J., & Carroll, J. (2005). A Virtual Reality Application for Stroke Patient Rehabilitation, *IEEE Int. Conf. on Mechatronics and Automation*, 1081-1086.
- Wikipedia,(2011). Canonical Form, retrieved 25 march 2012, from http://en.wikipedia.org/wiki/Canonical_form#Geometry
- Wu, Y., & Huang, T.S. (2001). Hand Modeling, Analysis, and Recognition, *IEEE Signal Processing Magazine*, Vol.18, Issue 3, May

APPENDICES

APPENDIX A: APPLICATION OF SOFTWARE

The program was written in Matlab R2009a, Calibration is done by clicking “Calibration” button and showing the hand to the camera just for few seconds. Both methods have two different working style based on user’s decision;

USE WEA: Use Wrist Extraction Algorithm suitable for the operator who wears short sleeved dress

NO WEA: Program pasts wrist extraction algorithm if the user wears long sleeved dress this results faster working. User can stop the program by pushing the “STOP” button.

The commands given by the user can also be read in “COMMAND” window in the interface.

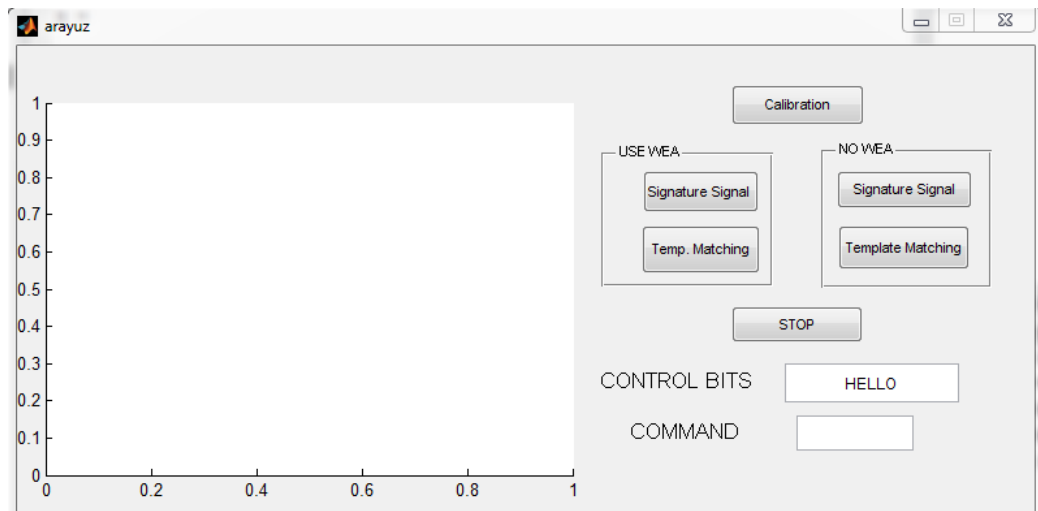


Figure A Results

APPENDIX B: SPECIFICATION OF SERVO MOTOR CONTROLLER

Runtime Selectable Baud rate. A serial message switches the baud rate from: 2400 to 38k4.
16 Servos. Servos driven simultaneously, continuously, 0-180° of rotation, 2uS resolution.
Servo Ramping. Choose one of 63 ramp rates for each servo.
Position Reporting. User may request position of an individual servo at any time.
Network Ready. Two modules may be linked together to drive 32 servos at the same time.
Power requirements: 5 VDC @ ~60mA for logic, 4.8 - 7.5 VDC for servos
Communication: Asynchronous Serial @ 2400ps or 38.4kbps (TTL or USB)
Dimensions: 57.14 x 45.74 x 16.53 mm
Operating temp range: 0°C to +70°C

Figure B Specification of Card

APPENDIX C: TEST RESULTS

Template Matching Algorithm Results

Shown Gesture	Orientation	Calculated Gesture
1	73,13352	1
1	68,77621	1
1	70,09566	1
1	74,14184	1
1	80,7269	1
1	-82,1484	1
1	-77,3339	1
1	-71,8358	1
1	-66,6297	1
1	-66,8573	1
1	-65,9777	1
1	-77,692	1
1	89,30041	1
1	74,79867	1
1	61,44768	1
1	53,2344	1
1	43,87721	1
1	38,95296	2
1	36,31317	2
1	50,85677	1
1	58,00033	1
1	66,71579	1
1	84,7439	1
1	-77,3655	1
1	-72,7523	1
1	-67,8728	1
1	-68,3267	1
1	-67,1437	1
1	-71,7406	1
1	-86,1105	1
1	86,35409	1
1	78,94329	1
1	67,30914	1
1	56,52618	1
1	50,04856	1
1	46,7638	2
1	45,67953	2
1	50,37373	1
1	67,95729	1

1	73,29085	1
1	75,137	1
1	78,70949	1
1	84,7544	1
1	-88,3847	1
1	-89,1771	1
1	-89,6744	1
1	-89,8677	1
1	-89,8641	1
1	89,99951	1
1	-82,9829	1
1	-73,1448	1
1	-69,2466	1
1	-66,7857	1
1	-65,9816	1
1	-66,2778	1
1	-74,2853	1
1	-81,6842	1
1	-86,0762	1
1	87,99187	1
1	78,01563	1
1	64,90961	1
1	55,59849	1
1	48,44531	1
1	42,38306	2
1	38,4705	2
1	38,11027	2
1	43,57506	1
1	49,29224	1
1	61,68688	1
1	68,50096	1
1	63,89272	1
1	51,34134	1
1	45,73948	1
1	40,65344	2

1	58,01924	1
1	80,65787	1
1	86,96854	1
1	-82,7703	1
1	-72,026	1
1	-69,9868	1
1	-63,5856	1
1	-63,5856	1
1	-63,8923	1
1	-77,0052	1
1	-81,6364	1
1	-86,6832	1
1	84,06266	1

1	71,75273	1
1	61,7933	1
1	51,3622	2
1	48,89119	2
1	48,04443	1
1	48,81085	1
1	67,31738	1
1	78,26486	1
1	64,73899	1
1	43,70274	1
1	40,49431	2
1	40,56076	2
1	41,3341	2

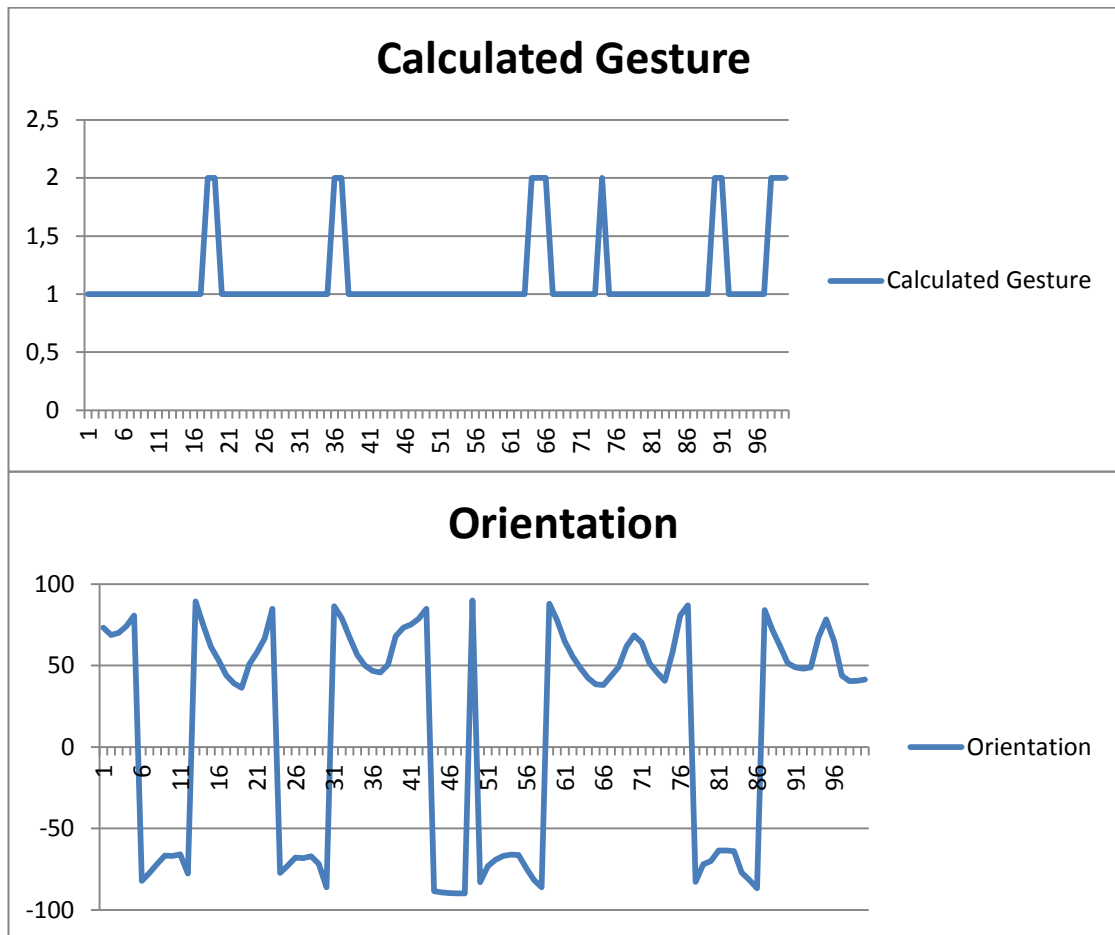


Figure C.1 Results for Gesture1 with TMA

Shown Gesture	Orientation	Calculated Gesture
2	74,68326	2
2	74,24544	2
2	75,12833	2
2	73,83849	2
2	68,39288	2
2	55,91124	2
2	49,96603	2
2	48,16428	2
2	42,13996	2
2	40,67618	2
2	40,39201	3
2	53,31823	2
2	66,24719	2
2	71,25881	2
2	79,58448	2
2	88,84005	2
2	-83,0212	2
2	-75,1497	2
2	-68,1073	2
2	-65,8426	2
2	-63,0742	2
2	-59,7204	2
2	-58,5606	2
2	-56,3037	2
2	-58,0086	2
2	-61,4801	2
2	-67,956	2
2	-73,8035	2
2	-78,2652	2
2	-82,508	2
2	-87,0444	2
2	89,45713	2
2	83,52414	2
2	78,5256	2
2	71,96332	2
2	64,94147	2
2	55,76778	2
2	50,74183	2
2	46,4342	2

2	46,77336	2
2	43,68332	3
2	42,27618	3
2	45,72569	2
2	52,11591	2
2	59,68582	2
2	67,54013	2
2	75,34714	2
2	82,66207	2
2	-86,2138	2
2	-80,1711	2
2	-73,3923	2
2	-68,7435	2
2	-62,7692	2
2	-51,2007	2
2	-52,2244	2
2	-77,0128	2
2	-86,8225	2
2	88,19099	2
2	81,91325	2
2	76,879	2
2	67,87291	2
2	56,28141	2
2	48,82553	2
2	46,09337	2
2	41,80803	3
2	39,97015	3
2	39,55809	3
2	51,47098	2
2	50,44143	2
2	63,81384	2
2	62,55587	2
2	48,72086	2
2	43,08779	2
2	39,85965	3
2	36,55814	3
2	37,38947	3
2	31,18488	2
2	34,17961	2
2	36,47177	2
2	36,13137	2

2	48,67783	2
2	65,23426	2
2	76,48235	2
2	86,42462	2
2	-80,135	2
2	-66,2068	2
2	-62,6199	2
2	-55,0745	2
2	-54,0733	2
2	-54,0874	2
2	-55,0032	2

2	-71,5858	2
2	-82,744	2
2	-87,9987	2
2	87,66654	2
2	81,8594	2
2	70,49973	2
2	60,75172	2
2	47,47087	2
2	40,33768	2

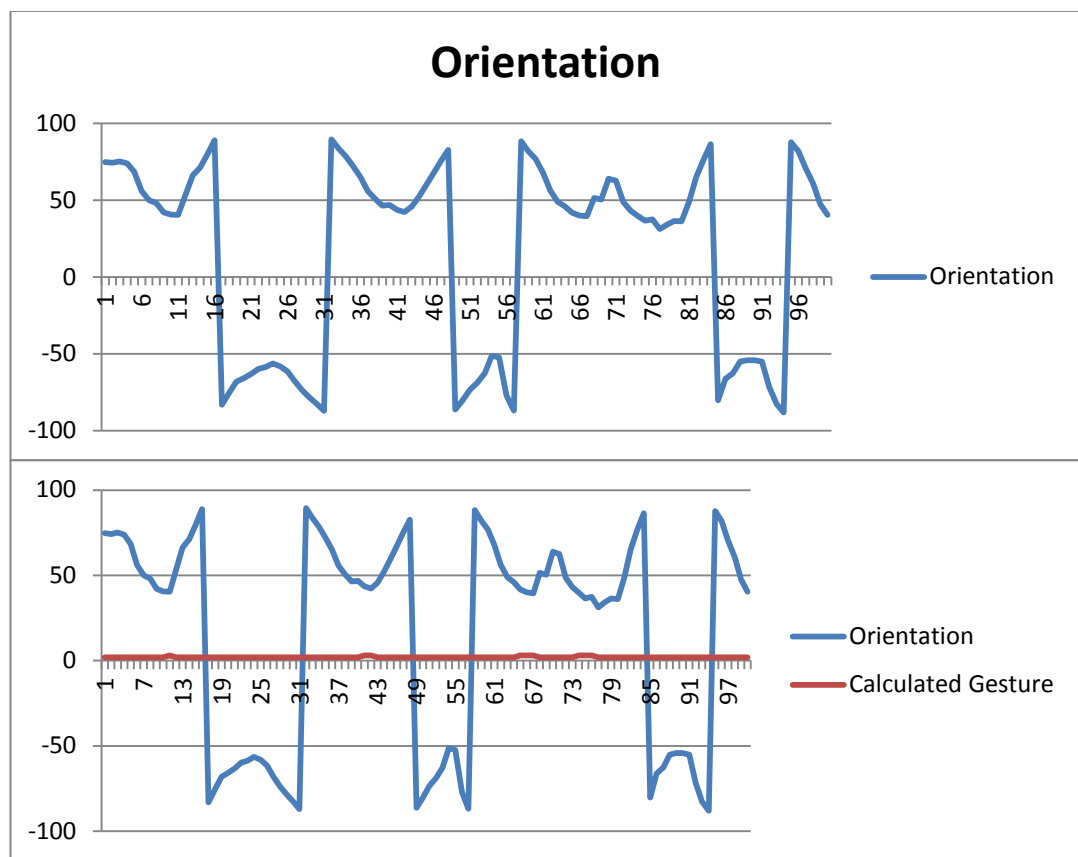


Figure C.2 Results for Gesture2 with TMA

Shown Gesture	Orientation	Calculated Gesture
3	82,79473	3
3	81,62545	3
3	81,6805	3
3	80,64789	3

3	73,25103	3
3	65,27485	3
3	52,69441	3
3	44,70704	3
3	47,6628	5
3	43,61083	3

3	65,97358	3
3	75,07994	3
3	80,61682	3
3	-89,9479	3
3	-82,4501	3
3	-72,0358	3
3	-68,6803	3
3	-62,2224	3
3	-59,3811	4
3	-58,8938	4
3	-71,9432	3
3	-80,4802	3
3	-84,7595	3
3	83,92303	3
3	78,38925	3
3	69,40807	3
3	65,39201	3
3	54,77352	3
3	49,05001	3
3	35,28925	3
3	34,12701	3
3	53,09126	3
3	66,17946	3
3	72,85966	3
3	86,02163	3
3	-84,1114	3
3	-78,9775	3
3	-76,9161	3
3	-63,7841	3
3	-53,2109	1
3	-46,6633	1
3	-56,73	3
3	-63,0909	3
3	-72,4058	3
3	-85,3669	3
3	-89,3466	3
3	81,91772	3
3	73,75259	3
3	63,66913	3
3	49,05707	3
3	37,63415	3
3	26,48124	3
3	28,52709	3

3	43,51934	3
3	60,50207	3
3	72,28982	3
3	87,19153	3
3	-81,2277	3
3	-69,6571	3
3	-61,3848	3
3	-48,105	3
3	-47,8294	3
3	-56,3997	3
3	-72,2463	3
3	-86,4524	3
3	89,03411	3
3	83,05849	3
3	72,43186	3
3	61,71284	3
3	51,08695	3
3	40,76185	3
3	28,99816	3
3	30,10773	3
3	38,43079	3
3	59,59134	3
3	70,52215	3
3	82,2464	3
3	-89,2464	3
3	-78,9767	3
3	-71,7648	3
3	-51,5299	3
3	-39,9857	3
3	-36,338	3
3	-40,4168	3
3	-64,7771	2
3	-76,8066	3
3	-85,6607	3
3	80,86574	3
3	71,19902	3
3	54,27626	3
3	32,18557	3
3	20,45673	3
3	38,15172	3
3	56,13868	3
3	71,68851	3
3	83,79928	3

3	-86,4897	3
3	-73,3258	3

3	-53,8794	3
3	-39,8257	3

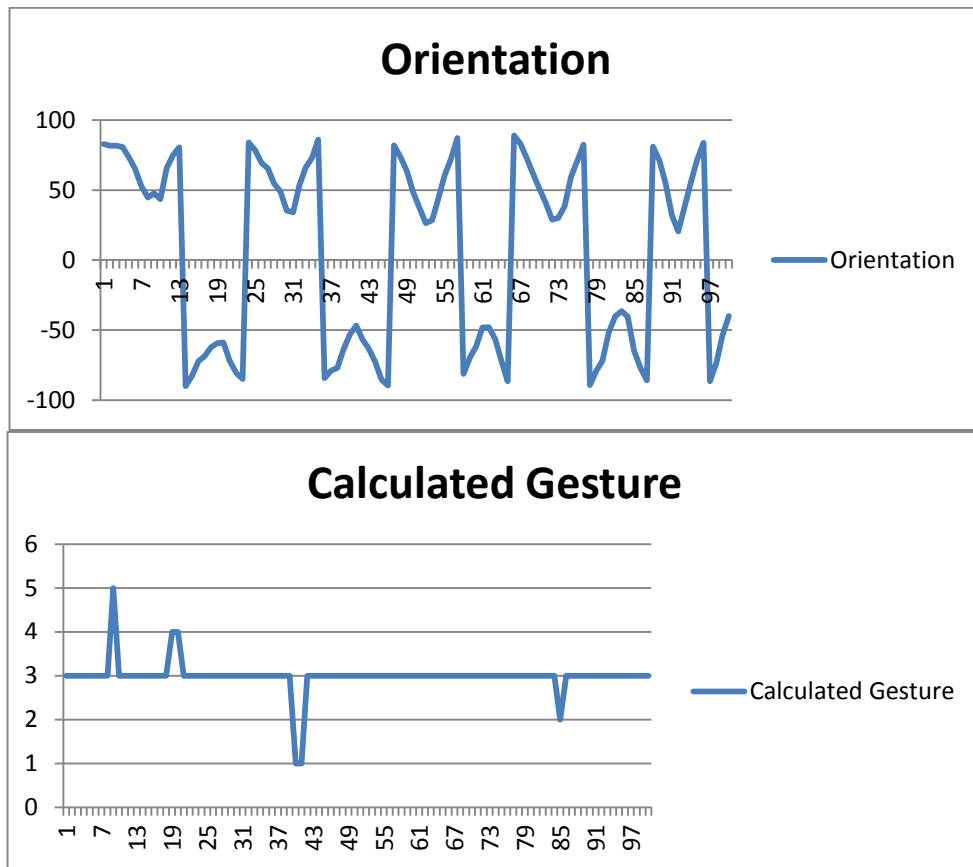


Figure C.3 Results for Gesture3 with TMA

Shown Gesture	Orientation	Calculated Gesture
4	78,72321	4
4	80,87468	4
4	80,92823	4
4	80,11569	4
4	77,316	4
4	70,09174	4
4	64,16058	4
4	57,84093	4
4	49,99976	4
4	49,34927	4
4	41,93365	4
4	38,41946	4
4	35,1938	4

4	33,40319	4
4	37,02345	4
4	50,7248	4
4	57,91105	4
4	66,42317	4
4	71,95696	4
4	80,52237	4
4	86,39528	4
4	-82,4585	4
4	-76,9238	4
4	-74,8599	4
4	-65,3309	4
4	-59,9314	4
4	-54,1997	4

4	-50,5956	4
4	-48,7951	4
4	-49,5879	4
4	-53,8644	4
4	-64,3332	4
4	-68,7501	4
4	-76,5219	4
4	-85,0393	4
4	88,49122	4
4	81,16152	4
4	68,98414	4
4	54,69332	4
4	49,98527	4
4	41,98084	5
4	40,3971	5
4	48,9247	4
4	67,83746	4
4	75,78707	4
4	86,72493	4
4	-87,6954	4
4	-78,3816	4
4	-70,5463	4
4	-63,5191	4
4	-56,9481	4
4	-52,3617	4
4	-50,4723	4
4	-46,0227	4
4	-65,6784	4
4	-74,0051	4
4	-82,8976	4
4	86,56782	4
4	71,22582	4
4	60,10121	4
4	48,917	4
4	34,78272	4
4	27,94028	4
4	32,99433	4

4	55,07182	4
4	68,62563	4
4	83,47523	4
4	-82,5474	4
4	-68,8027	4
4	-58,2126	4
4	-48,8081	4
4	-44,655	4
4	-51,9477	4
4	-64,685	4
4	-71,8877	4
4	-81,4547	4
4	85,92345	4
4	74,85413	4
4	59,7497	4
4	45,25351	4
4	29,42988	4
4	22,20524	4
4	29,18872	4
4	56,44777	4
4	67,73959	4
4	69,82356	4
4	88,79405	4
4	-63,7365	4
4	-59,7005	4
4	-48,0679	4
4	-46,5708	4
4	-59,3336	4
4	-72,9854	4
4	-85,6508	4
4	83,55553	4
4	68,36989	4
4	53,86048	4
4	38,0629	4
4	24,70363	4
4	28,02843	4

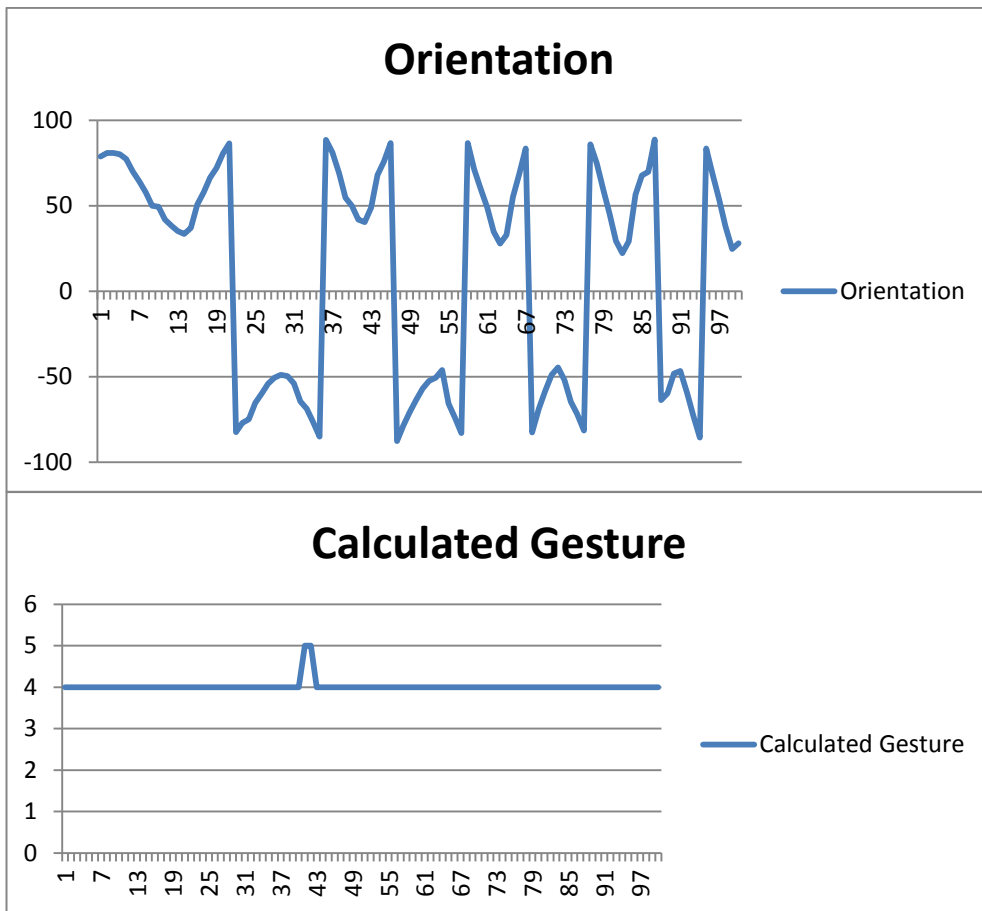


Figure C.4 Results for Gesture4 with TMA

Shown Gesture	Orientation	Calculated Gesture
5	82,23336	5
5	85,0236	5
5	85,57073	5
5	87,50176	5
5	-89,2456	5
5	-87,4062	5
5	-86,5652	5
5	-87,8232	5
5	-87,7924	5
5	-88,9898	5
5	-89,6617	5
5	89,39804	5
5	85,1953	5
5	75,40876	5
5	68,69815	5

5	63,16891	5
5	60,14907	5
5	54,77649	5
5	50,1563	5
5	47,87579	5
5	45,95099	5
5	42,60368	5
5	39,34063	5
5	39,32339	5
5	41,57119	5
5	57,28621	5
5	61,96042	5
5	72,62491	5
5	76,6813	5
5	83,98277	5
5	-86,7279	5
5	-83,6083	5

5	-78,6488	5
5	-74,8163	5
5	-70,5806	5
5	-66,3871	5
5	-61,6562	4
5	-59,0578	3
5	-57,908	5
5	-54,5592	5
5	-53,3341	5
5	-52,3463	5
5	-50,1407	5
5	-49,3674	4
5	-49,4336	4
5	-48,75	4
5	-47,7526	5
5	-43,3049	5
5	-42,8899	4
5	-42,8094	4
5	-40,0433	4
5	-40,1191	5
5	-39,9242	5
5	-37,3457	5
5	-37,5534	5
5	-38,8413	5
5	-38,2917	5
5	-38,8614	5
5	-38,7827	5
5	-42,3872	5
5	-47,5808	5
5	-48,4942	5
5	-50,2022	4
5	-55,3025	4
5	-61,4824	4
5	-68,6805	4
5	-76,748	5

5	-84,2341	4
5	-89,8484	5
5	88,02514	5
5	84,71591	5
5	80,41089	5
5	77,13857	5
5	70,46853	5
5	62,98452	5
5	53,77172	5
5	44,55462	5
5	38,10743	5
5	32,40145	5
5	26,31935	5
5	25,16074	5
5	30,53972	5
5	40,0887	5
5	47,04161	5
5	59,04811	5
5	69,39997	5
5	76,28306	5
5	87,85846	5
5	-86,9388	5
5	-81,035	5
5	-75,1812	5
5	-66,8885	5
5	-61,1129	5
5	-55,1747	5
5	-49,5935	5
5	-44,0093	5
5	-40,3756	5
5	-45,3048	5
5	-60,4282	5
5	-70,4201	5

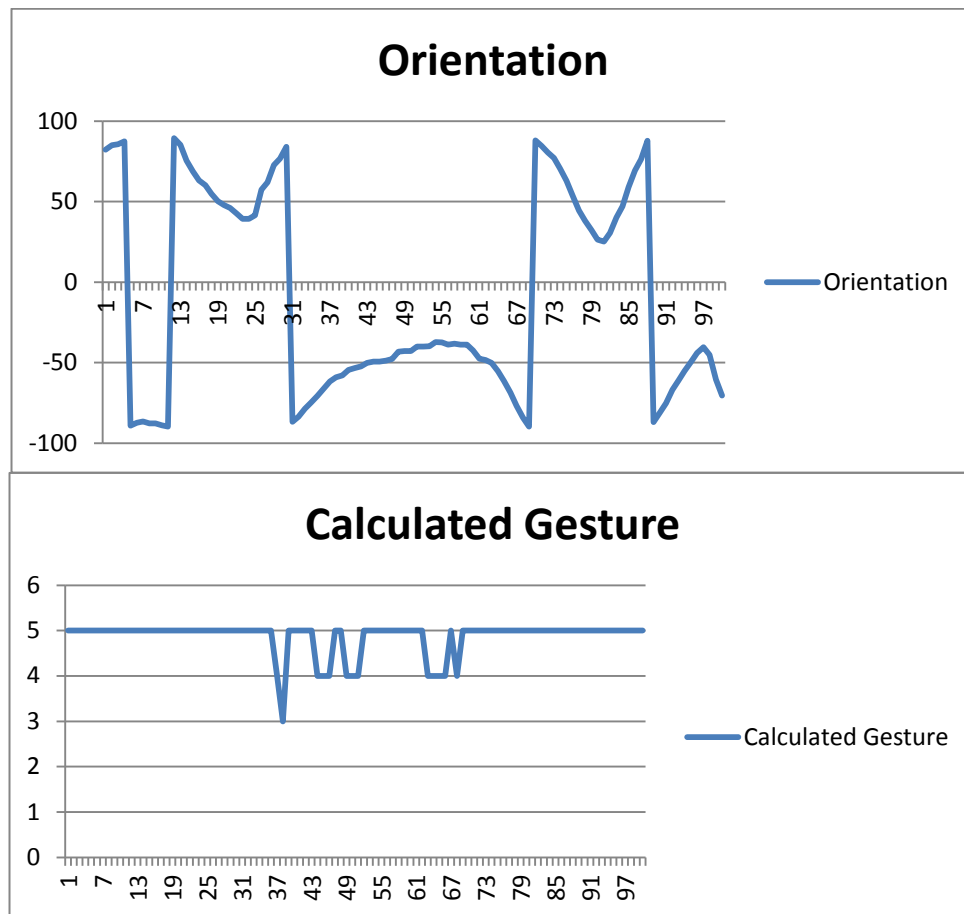


Figure C.5 Results for Gesture5 with TMA

Signature Signal Algorithm Results

Shown Gesture	Orientation	Calculated Gesture
1	86,53797	2
1	88,28592	3
1	85,85084	4
1	86,10408	2
1	-74,9478	2
1	-75,3693	2
1	-78,3478	2
1	-79,1695	1
1	-78,5193	1
1	87,3334	1
1	86,58022	1
1	85,78082	1

1	84,18643	1
1	83,3008	1
1	61,56938	2
1	56,0011	1
1	55,40096	2
1	53,91412	1
1	79,21728	1
1	-82,8511	1
1	-71,8732	1
1	-64,3441	1
1	-61,1992	2
1	-59,0984	1
1	-77,5969	1

1	-89,8712	1
1	87,83983	1
1	81,18291	2
1	70,33141	1
1	60,63319	1
1	50,98724	1
1	43,45447	1
1	43,18222	1
1	44,18436	1
1	53,04431	1
1	59,54985	1
1	65,20413	1
1	74,79482	1
1	83,00676	1
1	86,46675	1
1	-87,0207	1
1	-80,9287	1
1	-74,9006	1
1	-69,4217	1
1	-69,4217	1
1	-61,7127	1
1	-55,0804	1
1	-51,105	1
1	-56,8601	1
1	-57,1574	1
1	-62,9698	1
1	-69,1127	1
1	-73,2703	1
1	-76,0752	1
1	-84,0537	1
1	-89,7289	1
1	-86,3386	1
1	82,62005	1
1	75,26032	1
1	69,4953	1
1	63,64737	1
1	58,73562	2
1	50,99032	1

1	50,2728	1
1	56,16868	1
1	69,30746	2
1	82,8467	1
1	-88,0952	1
1	-78,4901	1
1	-69,0386	1
1	-58,7106	1
1	-52,2878	1
1	-51,4072	1
1	-61,5008	1
1	-63,8004	1
1	-70,4488	2
1	-78,953	1
1	-85,6505	1
1	86,59034	1
1	80,5896	1
1	74,33676	1
1	70,83191	1
1	69,16839	1
1	64,46471	1
1	59,19574	1
1	55,40417	1
1	48,48848	1
1	43,3621	2
1	52,92177	1
1	61,44209	1
1	72,55465	1
1	80,87028	1
1	-86,5055	1
1	-77,1514	2
1	-70,1987	1
1	-70,1987	1
1	-63,7926	1
1	-63,7926	1
1	-57,9889	1
1	-57,5359	1

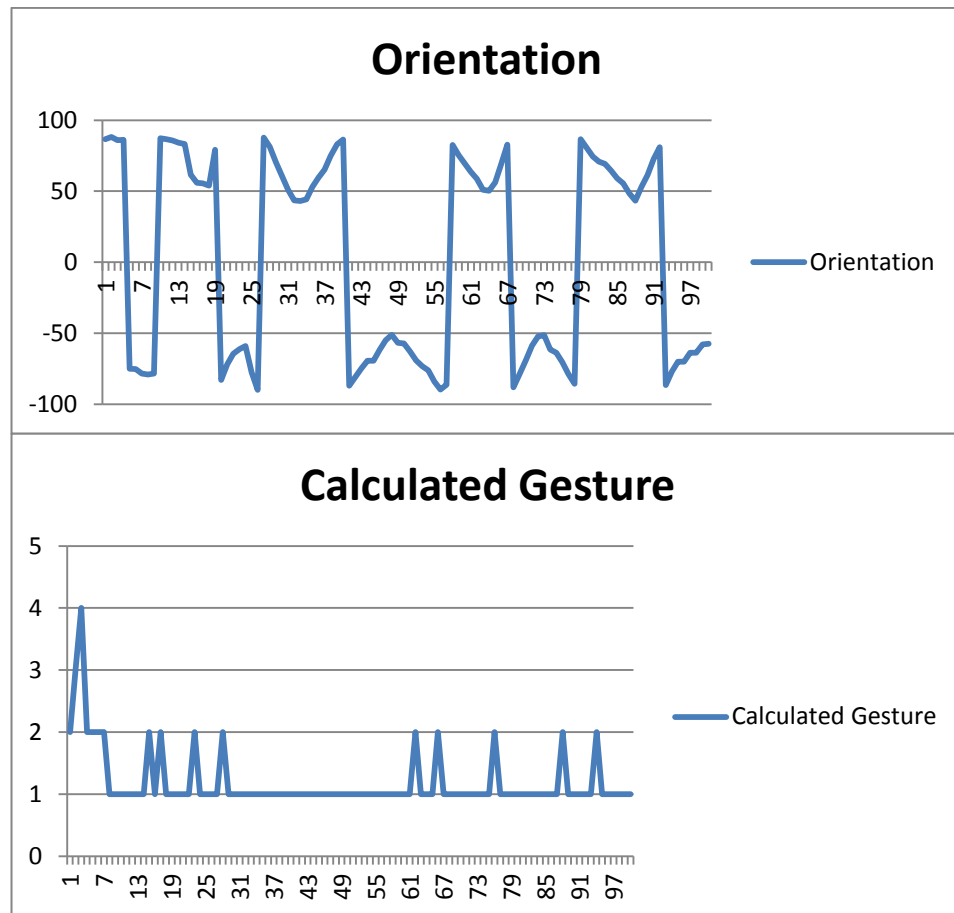


Figure C.6 Results for Gesture1 with SSA

Shown Gesture	Orientation	Calculated Gesture
2	72,9656	3
2	74,3333	2
2	73,29918	3
2	71,81189	3
2	67,51995	3
2	66,71411	2
2	61,34916	3
2	63,08594	2
2	60,55296	3
2	51,81649	3
2	42,78391	3
2	34,3023	3
2	64,7703	3
2	60,84842	2
2	62,05263	2

2	61,07806	2
2	48,56116	3
2	43,32956	3
2	31,63219	2
2	27,76817	2
2	32,5958	3
2	32,637	2
2	31,95852	3
2	29,87526	2
2	32,34956	2
2	33,52958	2
2	34,56582	2
2	34,04581	2
2	31,3221	2
2	42,98045	2
2	55,09939	2
2	72,15954	2

2	79,42454	2
2	-87,2157	2
2	-80,0903	2
2	-74,3108	2
2	-73,8988	2
2	-75,8101	2
2	-83,5143	2
2	82,19292	2
2	0	0
2	72,30484	2
2	63,40856	2
2	52,33574	2
2	45,34071	2
2	37,52128	2
2	33,08385	2
2	31,30172	3
2	31,61324	2
2	60,44343	2
2	72,81819	2
2	80,39376	2
2	86,15188	2
2	-86,4148	2
2	-76,9494	2
2	-71,7528	2
2	-72,2086	2
2	-73,6226	2
2	82,98778	2
2	75,93755	2
2	74,14906	2
2	67,16262	2
2	59,46949	2
2	49,54781	2
2	42,97454	2
2	37,76217	2
2	38,10524	2

2	36,02064	2
2	45,39987	2
2	55,87856	2
2	61,63588	2
2	65,69421	2
2	65,31177	2
2	63,92508	2
2	73,14114	2
2	78,46337	2
2	83,45495	2
2	84,60433	2
2	-86,9302	2
2	-79,8948	2
2	-71,0152	2
2	-61,5477	2
2	-62,6875	2
2	-77,2688	2
2	-88,0716	2
2	85,76888	2
2	77,71179	2
2	71,62023	2
2	66,33849	2
2	60,25801	2
2	54,65672	2
2	50,95444	2
2	43,78344	2
2	39,89413	2
2	36,31201	3
2	30,43021	2
2	28,33181	2
2	25,81731	2
2	40,93956	2
2	54,8423	2

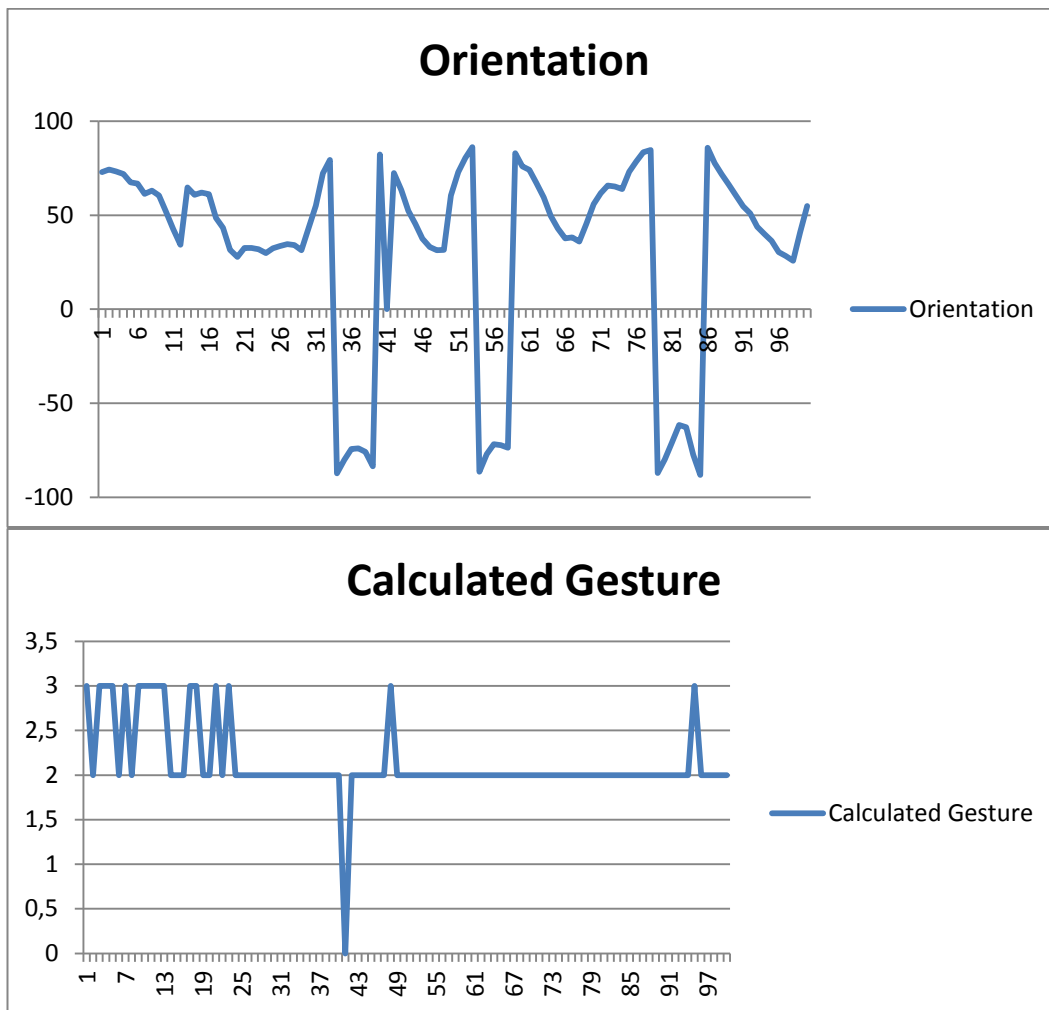


Figure C.7 Results for Gesture2 with SSA

Shown Gesture	Orientation	Calculated Gesture
3	80,66899	3
3	78,33371	3
3	77,50742	3
3	81,62777	3
3	-65,554	3
3	-40,2352	3
3	-30,5521	4
3	-29,8321	3
3	-32,3556	3
3	-35,6439	3
3	-41,3086	3

3	-66,2887	3
3	86,79143	3
3	74,06881	4
3	72,63306	4
3	50,40485	5
3	38,3663	4
3	74,95172	3
3	80,31204	3
3	79,75278	3
3	46,93352	3
3	38,47622	3
3	35,4455	3
3	35,21963	3

3	36,04407	3
3	39,42907	3
3	39,31301	3
3	39,86061	3
3	85,188	3
3	-74,8051	3
3	-66,0708	4
3	-39,461	3
3	-23,4212	3
3	-24,768	3
3	-26,2251	3
3	-27,9924	3
3	-36,9657	3
3	-47,6007	4
3	-48,6131	3
3	-78,3543	3
3	-88,3633	3
3	87,01643	3
3	73,71476	3
3	70,78868	3
3	59,44647	3
3	46,96257	3
3	38,26742	3
3	39,31383	3
3	57,13228	3
3	72,86069	3
3	-82,3578	3
3	-68,934	3
3	-56,9124	3
3	-47,1253	3
3	-37,1293	3
3	-31,9997	3
3	-42,1755	3
3	-58,7771	3
3	-78,1485	3
3	86,80167	3
3	74,60809	3
3	62,75384	3
3	52,3927	3
3	40,63603	3

3	29,54211	3
3	46,82922	3
3	65,20578	3
3	80,88981	3
3	-83,5747	3
3	-67,0015	3
3	-50,8296	3
3	-37,2172	3
3	-32,2342	3
3	-55,4373	3
3	-66,6799	3
3	-80,1158	3
3	84,79092	3
3	67,49654	3
3	54,56851	3
3	43,78744	3
3	36,14924	3
3	45,45013	3
3	72,08538	3
3	-86,314	3
3	-73,2469	3
3	-58,2866	3
3	-54,3634	3
3	-43,3029	3
3	-35,2803	3
3	-26,6644	3
3	-47,825	3
3	-71,6342	3
3	-85,5236	3
3	84,00583	3
3	71,60984	3
3	57,28974	3
3	46,20265	3
3	29,97849	3
3	31,45279	3
3	60,42987	3

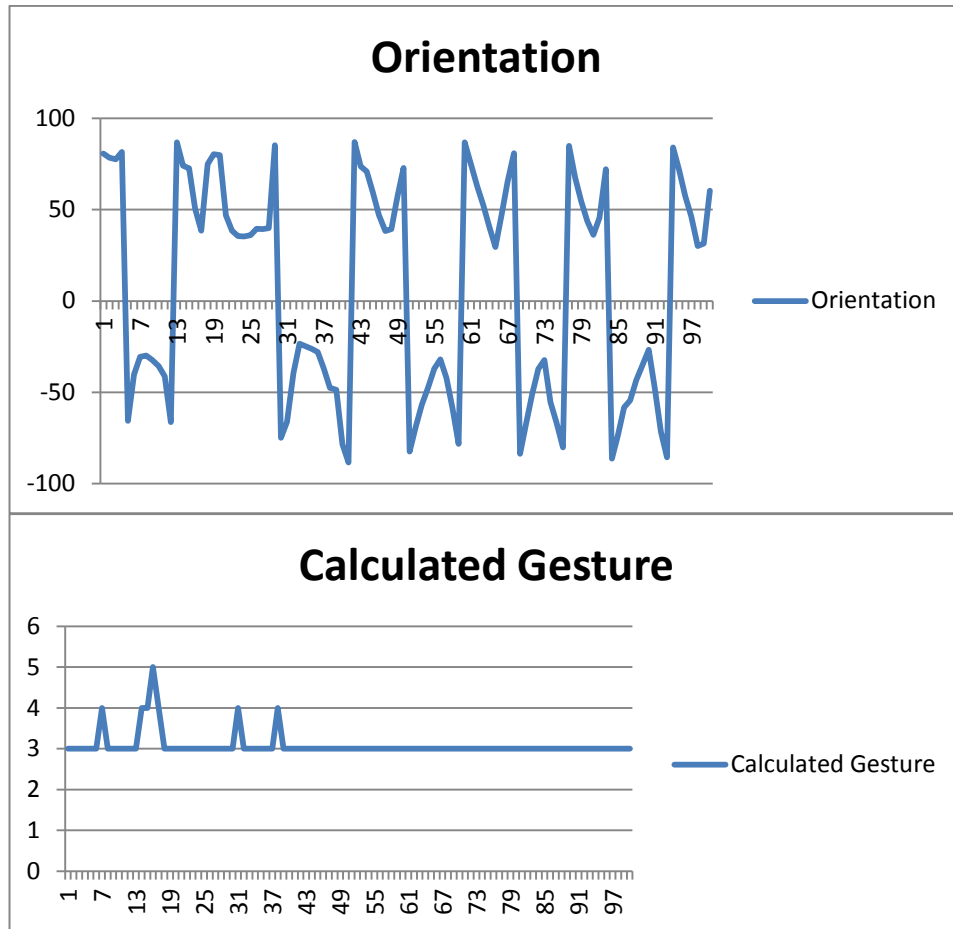


Figure C.8 Results for Gesture3 with SSA

Shown Gesture	Orientation	Calculated Gesture
4	84,86703	4
4	88,91936	4
4	89,12338	4
4	-89,1443	4
4	89,35973	4
4	89,28173	4
4	-76,1555	4
4	-58,7793	4
4	-43,9487	4
4	-34,9676	4
4	-32,2222	4
4	-24,546	4
4	-24,5632	4

4	-23,6363	4
4	-46,0155	4
4	-63,735	4
4	-75,6754	4
4	89,06739	4
4	73,483	4
4	53,33795	5
4	44,6991	4
4	37,84349	4
4	34,61865	4
4	35,44116	4
4	38,3619	4
4	66,51502	4
4	82,26766	4
4	86,08946	4

4	-88,2503	4
4	-78,4324	4
4	-62,15	4
4	-53,4756	4
4	-44,5674	4
4	-33,8947	4
4	-27,2906	4
4	-27,0591	4
4	-34,2034	4
4	-47,9168	4
4	-61,9193	4
4	-73,3514	4
4	-87,8794	4
4	84,9478	4
4	76,93311	4
4	71,60182	4
4	66,08904	4
4	60,29015	4
4	51,44774	4
4	42,76401	4
4	35,80423	4
4	29,40492	4
4	30,49219	5
4	64,38936	4
4	-83,8194	4
4	-81,0384	4
4	-60,8717	4
4	-51,7584	4
4	-46,6346	4
4	-39,8955	4
4	-37,5804	4
4	-36,3858	4
4	-35,6559	4
4	-59,332	4
4	-76,3172	4
4	-86,616	4
4	82,14303	4

4	68,68316	4
4	60,14012	4
4	42,92358	4
4	34,16685	4
4	26,73635	4
4	54,95109	4
4	87,45653	4
4	-70,5785	4
4	-55,6248	4
4	-47,2842	4
4	-34,9406	4
4	-29,1319	4
4	-41,6971	4
4	-77,1277	4
4	85,31175	4
4	58,47836	4
4	41,36952	4
4	31,64388	5
4	43,42468	4
4	-87,0592	4
4	-64,8493	4
4	-54,338	4
4	-38,3247	4
4	-24,9607	4
4	-28,489	4
4	-44,1033	4
4	-58,1855	4
4	-78,4544	4
4	-89,2155	4
4	76,03135	4
4	60,93422	4
4	47,61139	4
4	34,32005	5
4	30,99111	4
4	31,74589	4

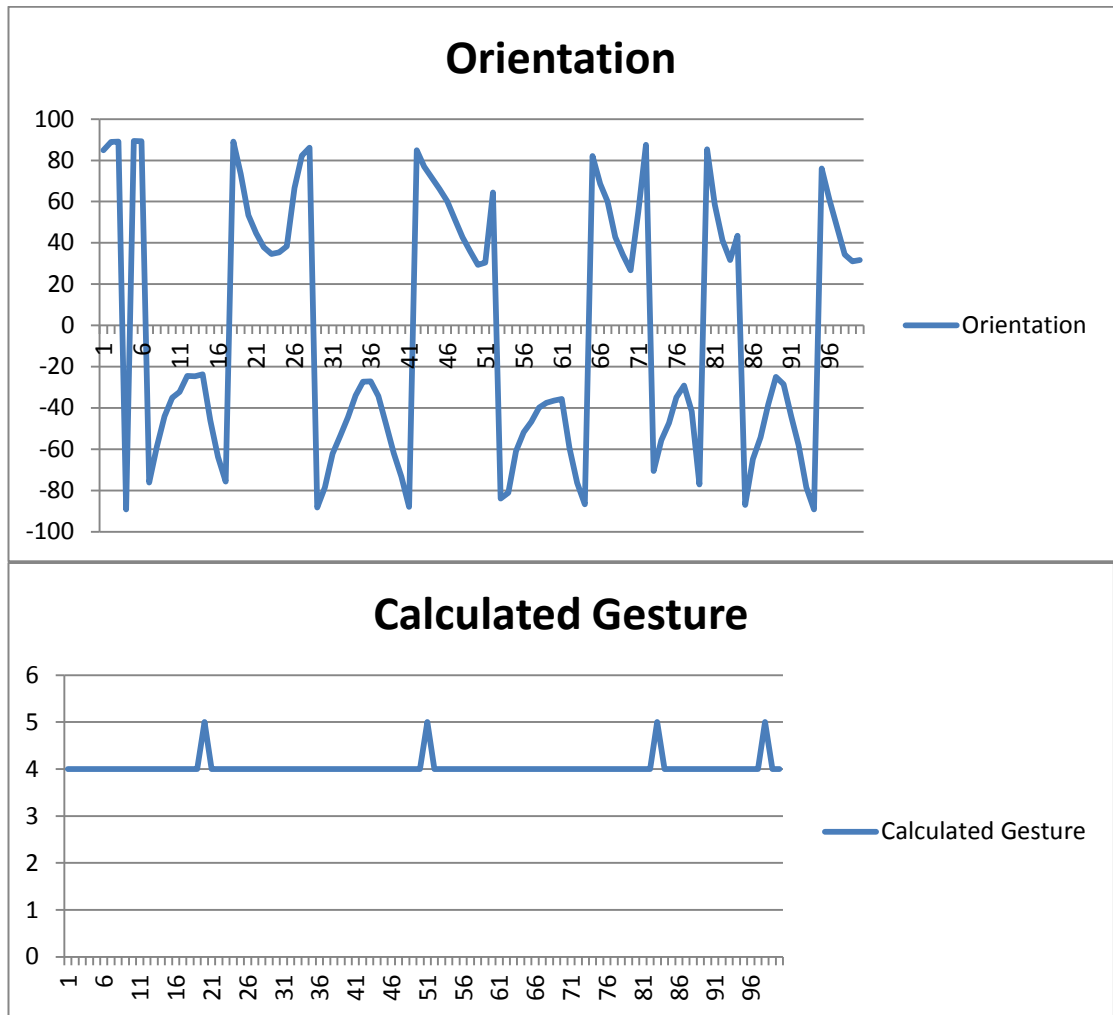


Figure C.9 Results for Gesture4 with SSA

Shown Gesture	Orientation	Calculated Gesture
5	79,19636	5
5	84,98765	5
5	83,55713	5
5	83,35272	5
5	72,26341	5
5	59,83427	5
5	51,39255	5
5	49,49574	5
5	74,85499	5
5	81,66113	5
5	89,38615	5

5	-83,748	5
5	-76,4523	4
5	-68,2952	5
5	-53,8467	5
5	-48,5108	5
5	-54,8461	5
5	-61,1451	5
5	-65,307	5
5	-81,2725	5
5	-88,3572	5
5	85,06	5
5	76,94381	5
5	70,95798	5

5	58,18444	5
5	47,84648	5
5	83,29482	5
5	-86,1728	5
5	-80,1312	5
5	-76,281	5
5	-57,9623	5
5	-54,6355	5
5	-80,8787	5
5	83,24909	5
5	79,53383	5
5	75,39336	5
5	73,08518	5
5	67,86772	5
5	64,42639	5
5	60,48564	5
5	54,3181	5
5	49,55487	5
5	70,23305	5
5	75,74227	5
5	81,4905	5
5	88,80178	5
5	-80,3437	5
5	-59,6856	6
5	-50,9485	4
5	-48,0294	5
5	-45,1384	5
5	-66,7645	4
5	-83,4807	5
5	82,45954	5
5	76,00825	5
5	71,17998	5
5	65,71917	5
5	61,52851	5
5	57,03204	5
5	50,36013	5
5	46,50937	5
5	47,31256	5
5	79,58983	5

5	88,78149	5
5	-82,9337	5
5	-75,5762	5
5	-67,6535	5
5	-59,3057	5
5	-54,0579	5
5	-65,4831	5
5	-76,0548	5
5	-80,79	5
5	-87,0967	5
5	88,2691	5
5	82,05793	5
5	77,54635	5
5	75,10953	5
5	73,08048	5
5	69,42097	5
5	59,29355	5
5	55,54776	5
5	52,16879	5
5	53,92159	4
5	69,51545	5
5	80,50633	5
5	86,17346	5
5	-84,2571	5
5	-81,0236	5
5	-76,043	5
5	-71,9068	5
5	-70,0489	5
5	-67,1958	5
5	-64,4748	4
5	-79,3878	5
5	88,87194	5
5	80,99343	5
5	71,98961	5
5	63,0335	5
5	57,89004	5
5	46,18583	5

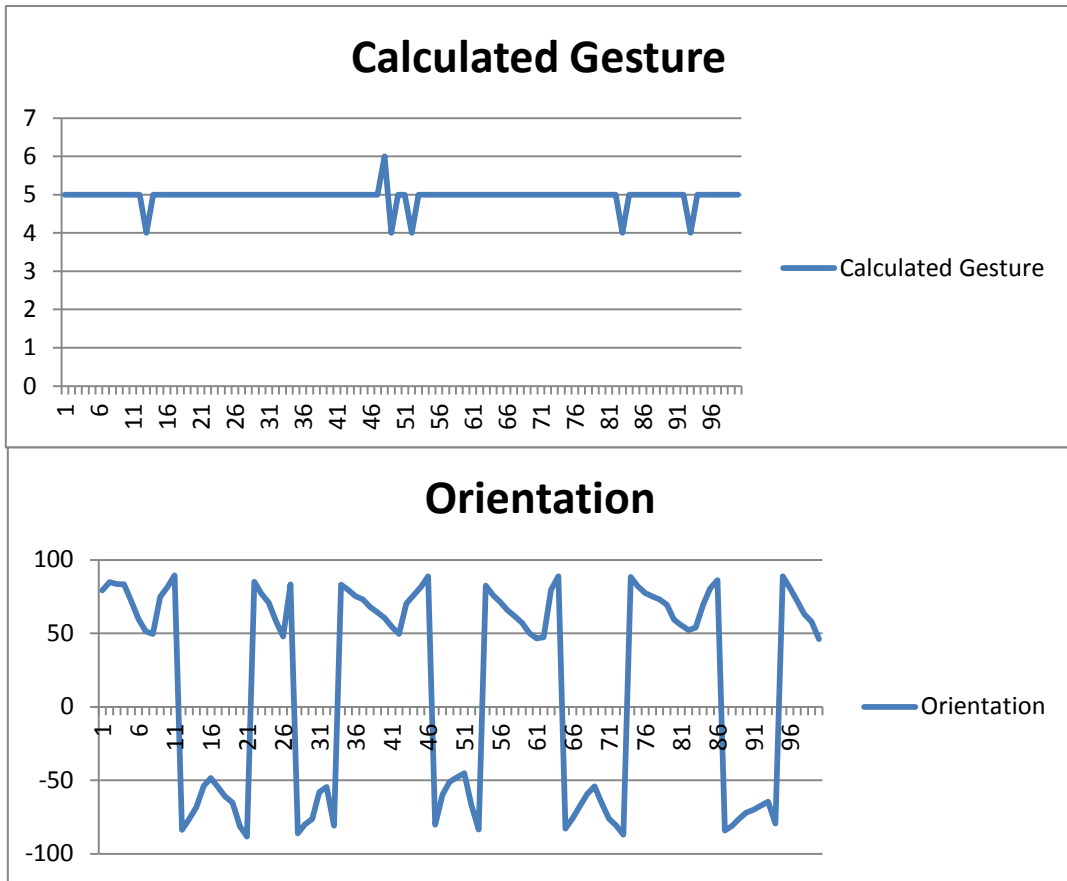


Figure C.10 Results for Gesture5 with SSA