A. C. MOGOL

ÇANKAYA UNIVERSITY

3D HAND RECONSTRUCTION WITH BINOCULAR VIEW

ALİ CAN MOGOL

DECEMBER, 2011

3D HAND RECONSTRUCTION WITH BINOCULAR VIEW

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ÇANKAYA UNIVERSITY

BY

ALİ CAN MOGOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

DECEMBER 2011

Title of the thesis: **3D Hand Reconstruction with Binocular View**

Submitted by : **Ali Can MOGOL**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Taner ALTUNOK

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Murat SARAN

Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Reza HASSANPOUR

Supervisor

Examination Date : 01. 12. 2011

Examining Committee Members

Prof. Dr. Mehmet R. TOLUN	(Çankaya Univ.)	
Asst. Prof. Dr. Reza HASSANPOUR	(Çankaya Univ.)	
Dr. Ersin ELBAŞI	(TÜBİTAK)	

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name	:	Ali Can MOGOL
Signature	:	
Date	:	01.12.2011

ABSTRACT

3D HAND RECONSTRUCTION WITH BINOCULAR VIEW

MOGOL, Ali Can

M.Sc., Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Reza Hassanpour

December 2011, 58 pages

In the field of Human Computer Interaction (HCI) one of the important goals is designing better interfaces for improving interactions between human beings and computers. There are lots of approaches to address this problem. One of these approaches is interfacing using human hand gestures. The capturing and modeling of the gestures and articulations of the human hand is a challenging problem. There exist hardware and software solutions proposed to solve this problem. In this thesis, an inexpensive fast and effective method to stereo capture and create 3D hand model is proposed.

The setup used for this thesis is; five different color markers and commodity hardware consists of two web cameras and a low cost laptop computer. In this thesis, starting with the stereo calibration of the cameras, capturing and tracking the color markers attached to the finger tips, leading 2D points of the finger tips, converting the 2D points to 3D points and calculating the finger articulations according to these 3D points and modeling the 3D hand have been accomplished, so the user can see his/her own hand's articulation on the screen as a 3D hand model.

Keywords: Human Computer Interaction, 3D Hand Model, Hand Gesture Detection, Stereo Calibration.

ÖZ

BİNOKÜLER GÖRÜŞ KULLANARAK BİR ELİN 3B YAPILANDIRILMASI

MOGOL, Ali Can

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Danışman: Y. Doç. Dr. Reza Hassanpour

Aralık 2011, 58 sayfa

İnsan bilgisayar etkileşimi alanında en önemli hedeflerden birisi de insan ve bilgisayar arasındaki etkileşiminin geliştirilmesi için daha iyi arayüzeyler tasarlanmasıdır. Bu problemin ele alınmasında bir çok yaklaşım bulunmaktadır. Bu yaklaşımlardan bir tanesi de insan eli hareketleri ile arayüzlemedir. İnsan eli hareketlerinin ve artikülasyonlarının yakalanması ve modellenmesi zorlu ve iddialı bir problemdir. Bu problemin çözülmesi için tasarlanmış donanım ve yazılımlar mevcuttur. Bu tez kapsamında, ucuz, hızlı ve etkili bir yöntem olarak stereo yakalama ve 3B el modeli oluşturma yöntemi önerilmiştir.

Bu tez için kullanılan düzenek, beş farklı renkteki renkli işaretler, kolaylıkla erişilebilir olan düşük maliyetli iki web kamerası ve bir dizüstü bilgisayardan oluşmaktadır. Bu tez kapsamında, stereo kalibrasyondan başlanarak, parmak uçlarındaki renkli işaretlerin yakalanması ve izlenmesi, parmak uçlarının 2B noktalarının bulunması ve bu 2B noktaların 3B'a dönüştürülmesi, ardından bu 3B noktalara göre parmak artikülasyonlarının hesaplanması ve sonunda 3B el modellenmesi yapılmıştır, böylece kullanıcı kendi elinin artikülasyonunu ekranda 3B el modeli olarak görebilmektedir.

Anahtar Kelimeler: İnsan Bilgisayar Etkileşimi, 3B El Modeli, El Hareketleri Yakalama, Stereo Kalibrasyon.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Asst. Prof. Dr. Reza Hassanpour for his guidance and support. I also thank him for giving me this opportunity to learn and grow as an engineer.

I also would like to thank to my wife Burçe, for her endless support and understanding duration of the thesis.

I also would like to thank to my family for their support along the years.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2-Dimension
3D	3-Dimension
CAMSHIFT	Continues Adaptive Mean Shift
CCD	Cyclic Coordinate Descent
CHI	Computer and Human Interaction
CMY	Cyan, Magenta, Yellow
CPU	Central Processing Unit
DifT	Difference Tolerance
DLS	Damped Least Squares
DOF	Degree of Freedom
FSM	Finite-State Machine
GCC	GNU Compiler Collection
GNU	Gnu is Not Unix
HCI	Human Computer Interaction
HMI	Human-Machine Interaction
HMM	Hidden Markov Models
HSB	Hue, Saturation, Brightness
HSI	Hue, Saturation, Intensity
HSV	Hue, Saturation, Value
IK	Inverse Kinematics
MMI	Man-Machine Interaction
OpenCV	Open Source Computer Vision
RGB	Red, Green, Blue
VTK	Visualization Tool Kit

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Today computers and other digital devices like gaming consoles, mobile devices, kiosks and interactive TVs, are in daily life more than ever. These devices have each different input devices and interfaces that not all of them easy to learn or intuitive. This theses concern with a generic interface for interaction with computers and digital device that is easier to learn and intuitive.

1.1.1 Human computer interaction

The Human Computer Interaction (HCI) term is used exchangeable with the terms “man-machine interaction” (MMI), “computer and human interaction” (CHI) and “human-machine interaction” (HMI). In these terms, the “machine” may be used for a wider meaning, most of them have the same meaning [1].

1.1.1.1 Definition of HCI

Although currently there is no agreed definition of the term, most of the time HCI has a broader meaning, but it may be simply define as “Human-computer interaction” (HCI) is the study of interaction between humans and computers [1]. Also it is defined as HCI is concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them [2].

1.1.1.2 Intuitiveness

The intuitiveness of an interface sometimes defined as usage of an interface by a human without training or rational thought. It is said that human beings "intuit" a concept when they seem to suddenly understand it without any apparent effort or previous exposure to the idea. However, it is clear that a user interface feature is "intuitive" insofar as it resembles or is identical to something the user has already learned. In short, "intuitive" in this context is an almost exact synonym of "familiar" [3]. According to this, the more the interface resembles to something that human beings are familiar, the more intuitive the interface.

1.1.2 HCI devices

There are many HCI interfaces for different devices. These interface devices are input devices like computer peripherals and control devices like industrial device interfaces.

1.1.2.1 Input devices

These input devices are computer peripherals or special hardware. They are used to send data to computers and other interactive/information systems like game consoles or medical equipments. Some of these input devices are well known devices like keyboard, mouse and joysticks and there are other less known devices like interactive pen displays and barcode readers. Some of these devices like mouse may have generic use and others like finger print reader may have only one use.

1.1.2.2 Control devices

There are specific interface devices for controlling various machinery and industrial devices. Sometimes they are called Human-machine interface (HMI) devices and they are local to one machine or equipment like construction machines and production line equipments. Although there are multipurpose control devices, most of these control devices are local to one machine like the controls of a crane.

1.1.3 Problems in HCI devices

HCI interfaces are generally designed for a specific device or a domain of devices. This leads to dissimilar interfaces for different devices, and most of these interfaces require special training and some time to master them for the operator of that device, and they may include expensive hardware parts like construction equipment interfaces and process control systems [4].And there are interfaces that are bounded by the infrastructure that they are build on, which the infrastructure is the limiting factor for the interfaces usability [5].

1.2 Aim & Scope

The aim of this thesis is; there is an easy, fast, inexpensive, effective way available, using commodity hardware and free software with minimum hardware requirements.

1.2.1 Hardware and software aspects

As in hardware, a setup formed of the minimum amount of hardware and funds. In the setup, there are two (2) low-end web cameras and a computer like a laptop. Hardware perspective is easy to implement and easy to port the solution to a lot of industries.

As in software, our aim is to capture, track and find the 2D points of the color markers, translate these 2D points to 3D points, find the articulations and reconstruct 3D hand to show the articulation of a human hand.

1.3 Method

In this thesis setup two roughly aligned cameras are used to capture the images. These images are undistorted, rectified and row-aligned. After rectification, a section found as the working section of the images. In the working section, the correspondence points are found with disparity information. Using this disparity information, the distance of a point in this resulting image is calculated by triangulation [6].

Five different color markers are used in this thesis and these markers are attached to the fingertips. These color markers are detected in the captured images. Images captured from cameras are in the RGB (red, green, blue) color model by default. The RGB values change with the amount of the light so the images are transformed to the HSV (hue, saturation, value) color model. In the HSV color model, not the hue and saturation values but only the V (value) changes with the amount of light [7]. Color markers in each image found by setting the hue and saturation values (ranges) for each color marker and finding the corresponding pixels.

Each color marker's center point is calculated as 2D point in the images of the left and right cameras, and using the disparity information, these points converted to 3D points.

These 3D points are used as the fingertips of the reconstructed 3D human hand. The articulations of the fingers are considered in two groups. First group is the thumb's articulations and the second group is the other four fingers' articulations. For some of the 3D points, there is more than one solution for the thumb's articulation, because of this reason; the outer-most solution is accepted, considering a faster response time and less CPU intensive calculations. The other four fingers' articulations are similar to each other, and this articulation is calculated according to relationship of the joint angles of the finger and the 3D point [8].

1.4 Results

In the scope of this thesis, below steps are successfully completed and each step has its results.

1.4.1 Calibration

At this step roughly aligned two cameras are calibrated, rectified and row aligned, resulting important values to calculate the 3D points, like the disparity map and reprojection matrix.

1.4.2 Capturing

Acquiring images from two web cameras, converting the RGB to HSV color model, finding the color markers' done in this step, resulting the 2D center points of each color marker.

1.4.3 Finding 3D position

Calculations using reprojection matrix and 2D points of each color marker done in this step resulting the 3D points of each color marker.

1.4.4 Articulation

According to the each 3D point and if the finger is the thumb or one of the other four fingers, articulations calculated, resulting a hand model with 3D finger articulations.

As the result of this thesis, using five color markers and two cameras, a 3D hand articulation is accomplished.

CHAPTER 2

BACKGROUND

2.1 Basic Concepts / Methods

There are basic concepts concerning the technologies and the techniques that are used in this thesis. These are the tools to develop the software and the methods to capture, track and model the 3D hand.

2.1.1 Technologies

There are different technologies involved in this thesis. These are GNU/Linux [9] as the operating system, GNU Compiler Collection (GCC) [10] as the compiler for the C / C++ source code, Open Source Computer Vision (OpenCV) [11] as the library of functions for computer vision, Visualization ToolKit (VTK) [12] for the 3D graphics and visualization.

2.1.2 Stereo imaging

For a two cameras setup, stereo imaging done in four steps [6]. The calibration is done by a calibration object. There are calibration objects like a cube or a chessboard pattern. A chessboard can be used as a calibration object, is shown in Figure2.1. It is because the flat chessboard patterns are much easier to deal with. Capturing different orientation images of the chessboard provides information to solve the locations of these images in global coordinates, relative to camera, and the camera intrinsic [13].

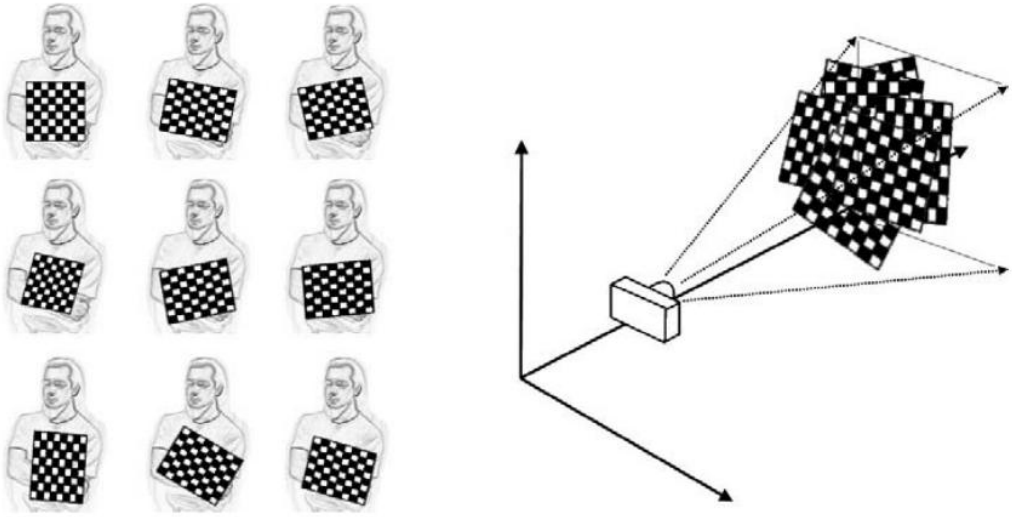

Figure 2.1: A Chessboard as a Calibration Object [6]

Images of a chessboard being held at various orientations (left) provide enough information to completely solve for the locations of those images in global coordinates (relative to the camera) and the camera intrinsic.

The result of stereo imaging step will be used to convert the 2D points to 3D points.

2.1.2.1 Undistortion

The cameras capture the raw images and these images are distorted. The raw images are shown in Figure 2.2.

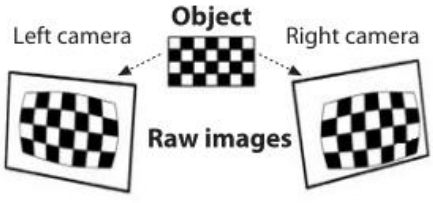

Figure 2.2: Raw Image Acquired from Cameras [6]

There are mainly two kinds of distortions coming from cameras. These two distortions are well known for the cheap cameras like web cameras. First one is the “Radial Distortion” which is a result of the shape of the lens, is shown in the Figure 2.3.

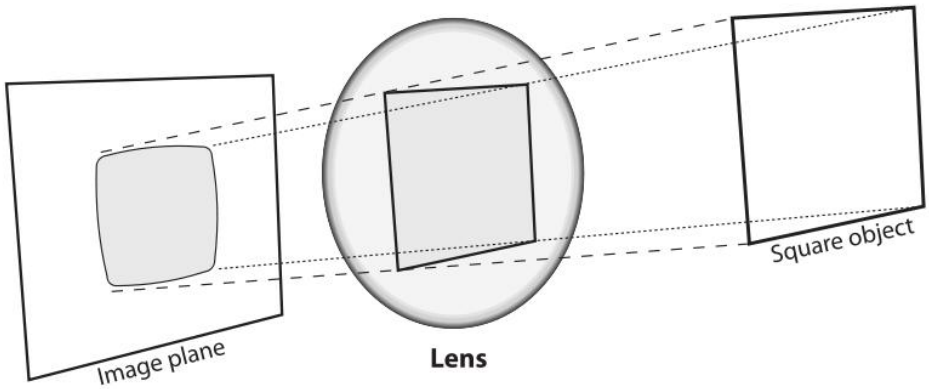

Figure 2.3: Radial Distortion as a Result of the Shape of the Lens [6]

The second one is the "Tangential Distortion" that arises from the assembly process of the camera [13], shown in the Figure 2.4.

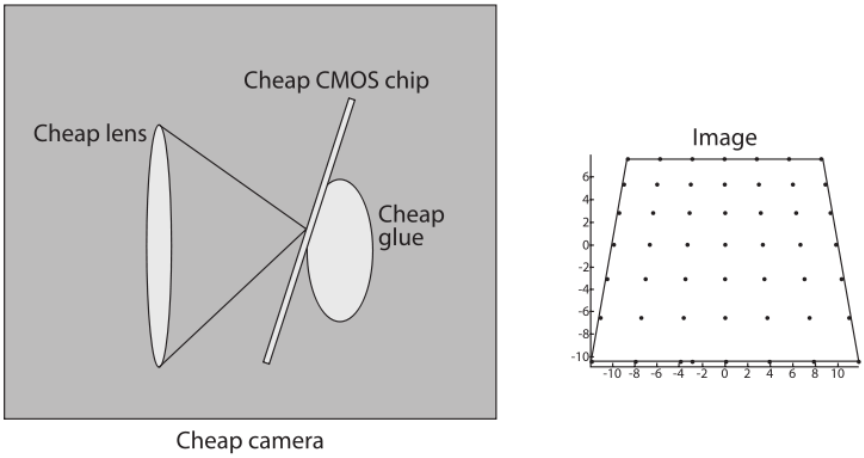

Figure 2.4: Tangential Distortion Because of Imperfect Assembly Process of the Camera [6]

After undistortion step, the image is an undistorted image [6], shown in Figure 2.5.

Figure 2.5: Undistorted Images [6]

2.1.2.2 Rectification

The angles and distances are adjusted between the two cameras and the result of that is row aligned and rectified images [6]. This means that the left and right images are coplanar and a feature at the left image is at the same row (x-coordinate) at the right image, shown in Figure 2.6.

Figure 2.6: Rectified, Row Aligned Images [6]

2.1.2.3 Correspondence

The two cameras are now undistorted and row aligned, so finding same features in the left and right images done in this step, resulting a disparity map. The differences in the x coordinates of the same features, coming from left and right cameras, x^l-x^r are the disparities [6].

2.1.2.4 Reprojection

Knowing the alignment of the cameras, disparity map is turned into the distance by triangulation; the result is a depth map [6].

A 3D point can be projected into one of the images coming from left or right camera. This can be done by a matrix multiplication in homogenous coordinates, using the projection matrix.

$$P = \begin{pmatrix} F_x & 0 & C_x & -F_x T_x \\ 0 & F_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.1}$$

In the above equation P is the projection matrix, and the F_x and F_y are the focal lengths of the rectified images. The C_x and C_y are the optical centers and the T_x is the translation of the camera relative to the left camera.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.2}$$

On the left there is a 3D point representation by homogenous coordinates, and on the right, it is the projection using matrix multiplication. The depth of a feature can be calculated from the image coordinates with the *reprojection matrix*, if the points in the left and right images correspond to the same scene feature.

$$Q = \begin{pmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 0 & F_x \\ 0 & 0 & -1/T_x & (C_x - C_{x'})/T_x \end{pmatrix} \tag{2.3}$$

In the above equation Q is the *reprojection matrix*, the primed parameters are from the left projection matrix, the unprimed from the right. If there are two matched points in the left and right image, like (*x*, *y*) and (*x'*, *y*) with a $d = x\text{-}x'$, then

$$\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = Q \begin{pmatrix} x \\ y \\ d \\ 1 \end{pmatrix} \tag{2.4}$$

where (X/W, Y/W, Z/W) are the coordinates of this feature and the *d* is the disparity. Assuming $Cx = C'x$, the distance (Z) can be calculated using triangulation like below,

$$Z = \frac{F - x\ Tx}{d} \tag{2.5}$$

Since the images are rectified, this reprojection is valid for these images [14].

2.1.3 Capturing and tracking

To correctly capture the color markers, it is important to choose the color space and the algorithms for the tracking.

2.1.3.1 Color space

Today there are color models (also called color space) either towards hardware or toward application. In the image processing context, RGB(red, green, blue) is the model for broad class of video cameras, the HSI (hue, saturation, intensity) model corresponds to closely with the humans perception of light. The HSI model also has the advantage that it decouples the color and gray-scale information in an image [15]. There are some variants of HSI systems, such as HSB (hue–saturation–brightness), HSL (hue–saturation–lightness), and HSV (hue–saturation–value) [16] [17] [18].

2.1.3.1.1 Conversion from RGB to HSV

8-bit RGB values are used for the source image. The conversion of the 8 bit RGB values to HSV values:

$$V \leftarrow \max(R, G, B) \tag{2.6}$$

$$S \leftarrow \begin{cases} \frac{V - \min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

$$H \leftarrow \begin{cases} 60(G - B)/S & \text{if } V = R \\ 120 + 60(B - R)/S & \text{if } V = G \\ 240 + 60(R - G)/S & \text{if } V = B \end{cases} \tag{2.8}$$

if $H < 0$ then $H \leftarrow H + 360$

On output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$

Since out images are 8-bit images, the HSV values are in the ranges; $V \leftarrow 255V, S \leftarrow 255S, H \leftarrow H/2$ (to fit to 0 to 255)

So the V can be removed, and ranges can be set for saturation and histograms for hue for each color marker.

2.1.3.1.2 Hue – saturation value

In this thesis, the HSV color model is used because of the RGB values changes with the amount of light, but in the HSV color model only the V (value) changes with the amount of light and H (hue) and S (saturation) values do not [7].

The RGB is represented by a cube, with the positive 0-1 range values in the three axes, shown in the Figure 2.7.At the origin (0,0,0) the color is black, for the maximum values of the cube at point (1, 1, 1) the color is white, from origin to this point is the gray line. At the maximum values of x, y and z axes there are

Red, Green and Blue maximum values. And at the last three corners there are the colors of cyan (C), magenta (M) and yellow (Y).

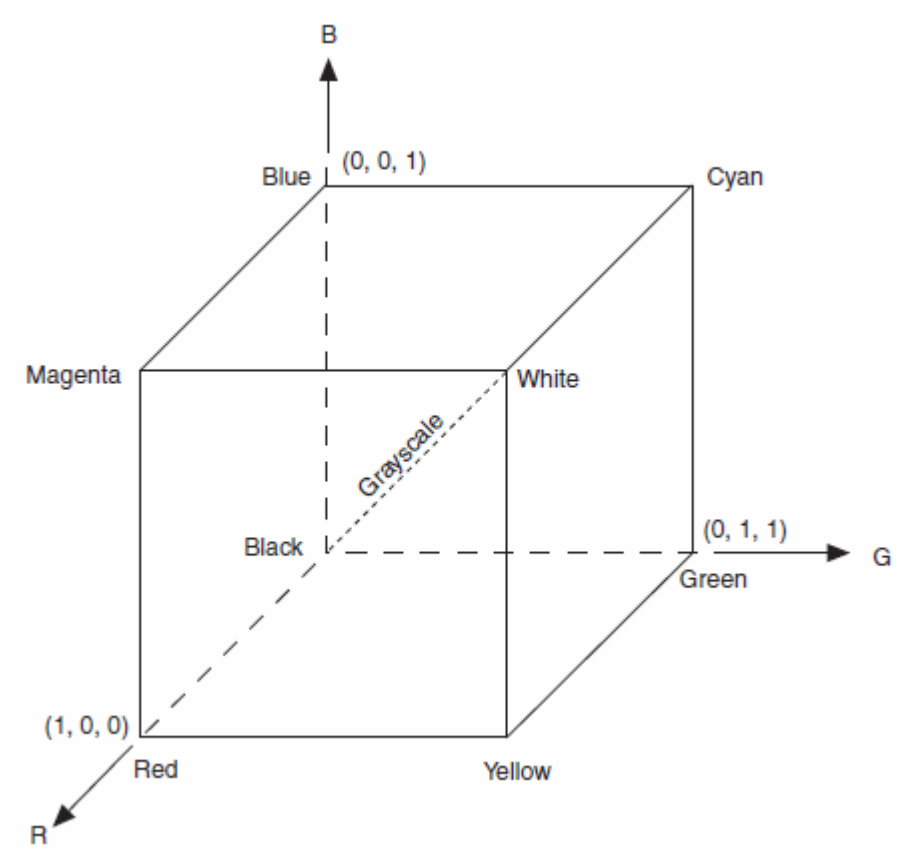

Figure 2.7: RGB Color Space's Representation as a Cube

Tilting the cube on the black to white axis and forcing RGBCMY to one plane at the white color's level, produces the HSV "hexcone" model, shown in Figure 2.8.

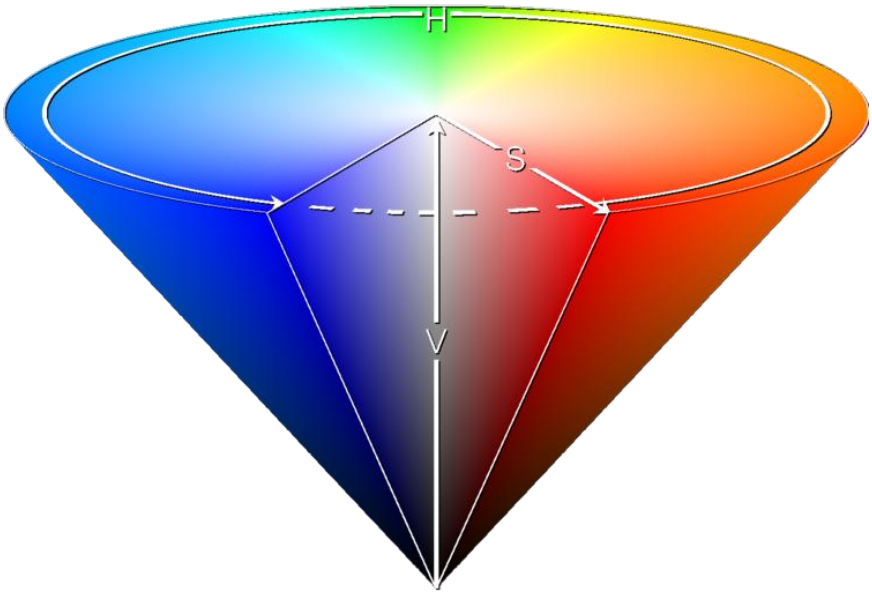

Figure 2.8: HSV Color Space as a "Hexcone"

In this color space the hue is the rotation and the saturation is the distance from the center, but the Value is the distance from bottom to top. For the HSV, it is most of the time enough to describe a color with proper hue and saturation values or some ranges.

2.1.3.2 Tracking

Tracking an object has many aspects like finding the object in each frame and removing the noise if possible.

2.1.3.2.1 CAMSHIFT

Continues Adaptive Mean Shift (CAMSHIFT) algorithm is a modification of the mean shift algorithm. Mean shift is an algorithm to find the mode (peak) of probability distributions. For each video frame, the probability distributions change, therefore mean shift algorithm modified to deal with these changes and the resulting algorithm is called the CAMSHIFT algorithm [19], this is shown in the Figure 2.9.

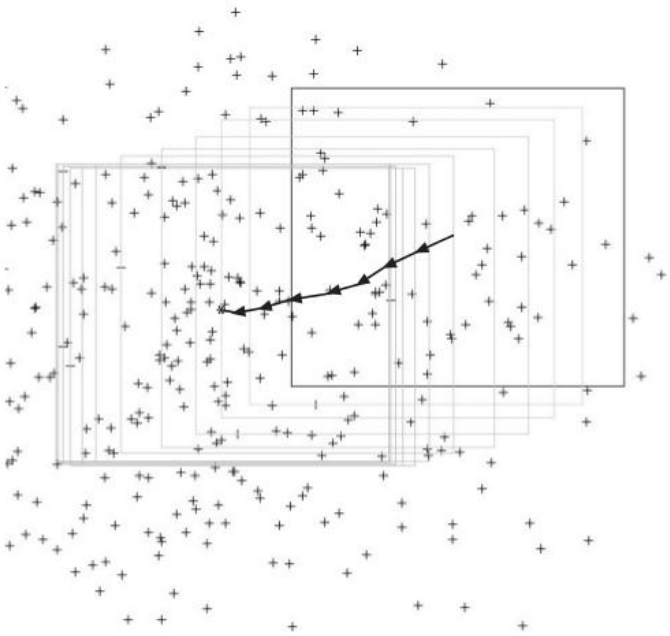

Figure 2.9: Continues Adaptive Mean Shift Algorithm Result in Each Frame [6]

2.1.3.2.2 Forward neighbor search

In pixel connectivity, for the 2D connectivity there are 4-connected (Von Neumann neighborhood) and 8-connected (Moore neighborhood) connections. This algorithm focuses only to the three (3) neighbors of a pixel. For an eight (8) connected pixel, these three neighbors are South, Southeast and East pixels, these pixels and the search algorithm shown in the Figure 2.10. This algorithm starts with the first pixel that has the values of acceptable hue and saturation, and continues to its neighbor pixels, until there is no other neighboring pixel acceptable. There may be a minimum limit that defines the distance as in pixels, between starting and ending pixels. If the ending pixel that is found at the lower right, found at a distance that is less then this minimum limit, those pixels that in the area bounded by the starting pixel and ending pixel are discarded. Setting this minimum limit to a value removes the noise and helps the CAMSHIFT algorithm to work better.

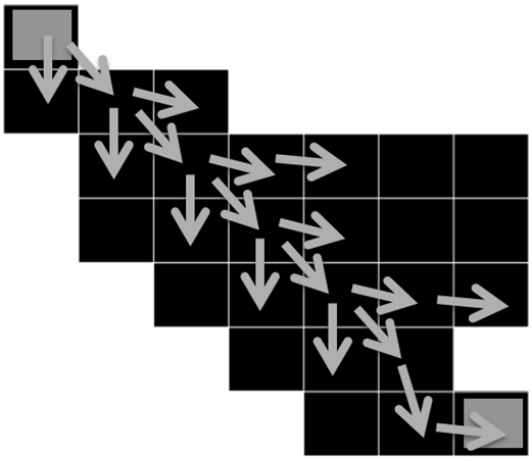

Figure 2.10: Forward Neighbor Search Algorithm Searching South, Southeast and East Neighbor Pixels

2.1.4 Articulations and kinematics

Finger articulations are considered in two groups. Because the thumb and the other four fingers articulations are different. For the thumb articulation, there are some 3D points that have more than one solution, but for the other four fingers there is one solution for each point.

2.1.4.1 Thumb articulation

The thumb articulations are complex and for some points there are multiple solutions. For a faster response time and less CPU intensive calculations, the outer-most solution of the thumb articulation is accepted; this solution is shown in the Figure 2.11.

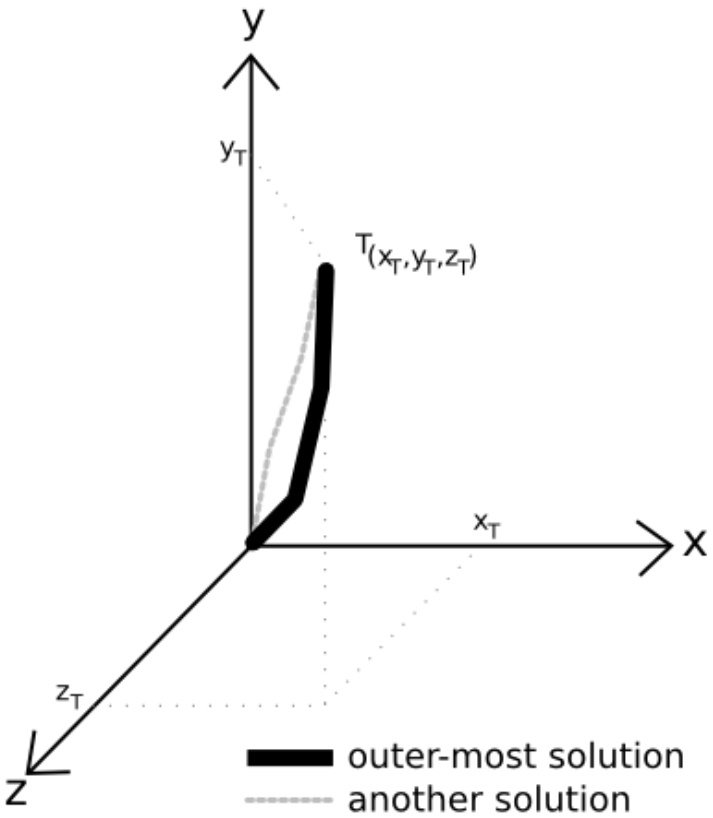

Figure 2.11: The Outer-Most Solution of the Thumb Articulation

2.1.4.2 Index to little fingers articulation

For these four fingers, there are three angles of a finger. The first angle is the angle between finger and the hand at the base joint, the second one is the middle one between the middle and inner phalanx (β angle), and the third one is the angle between the outer-most and the middle phalanx (α angle), these angles are shown in the Figure 2.12.

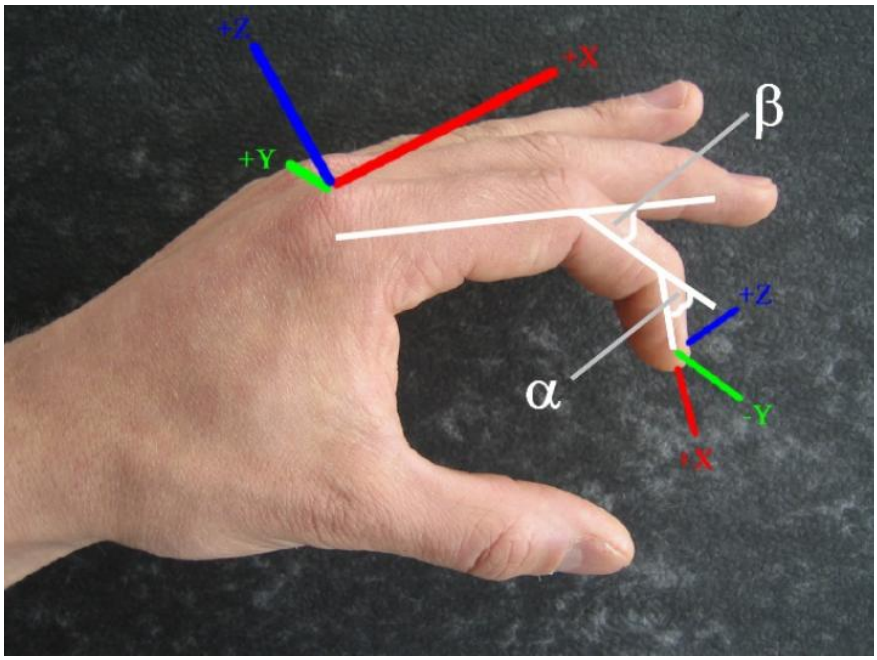

Figure 2.12: The Angle between the Middle and Inner Phalanx (β Angle), and the Angle between the Outer-Most and the Middle Phalanx (α Angle). [8]

There is a relationship between α and β angles, and it is approximated by a quadratic equation that gives the ratio of α / β.

$$q_{\alpha,\beta} = 0.23 + 1.73d + 1.5d^2 \qquad (2.9)$$

In this equation the d is the distance between the base joint and the fingertip relative to the maximum distance [8].

For a known 3D point, the articulation of the finger can be calculated iteratively according to this equation.

2.2 Similar Systems

There are similar systems that resembles to the system that is proposed in this thesis. These systems may be based on hardware or software.

2.2.1 Hardware systems

These systems are used for 3D hand reconstruction or capturing the position and shape of a hand. There are many transmitters; like infrared optical finger trackers [8], wireless acoustic emitters [20] and gloves are involved [21].

The transmitter approach works with a transmitter at the fixed positions of the tracking object, a device with transmitters shown in the Figure 2.13. In this case the objects are hand, fingers or joints of the fingers. Using transmitters and a receiver for these transmitters, the position of the transmitter or the distance can be calculated.

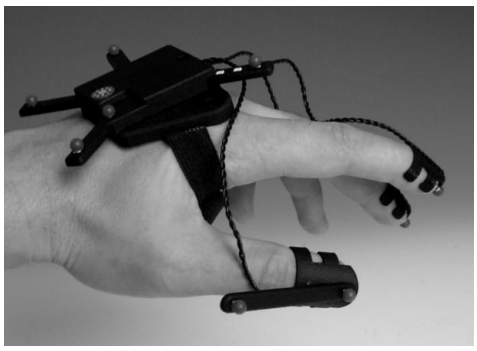

Figure 2.13: Transmitters Attached at the Finger Tips [8]

The wearable devices like gloves are used to track the position, movement and articulations, as shown in the Figure 2.14. These kinds of devices may have sensors to capture pressure, proximity and resistance, and they may have the processing unit (a portable/wearable computer) with them.

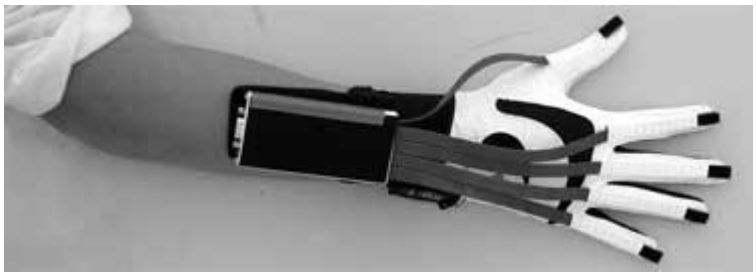

Figure 2.14: A Wearable Glove to Track the Position, Movement and Articulations of Hand [8]

2.2.2 Software systems

There are software systems that do not involve any hardware as in the transmitters or electronics at the hand side. These systems capture images from camera(s) and process them according to their own approaches.

2.2.2.1 Learning – training – fitting systems

These systems capture images of a hand; most of the time there is nothing attached to the hand or fingers, and they capture the hand using the skin color segmentation.

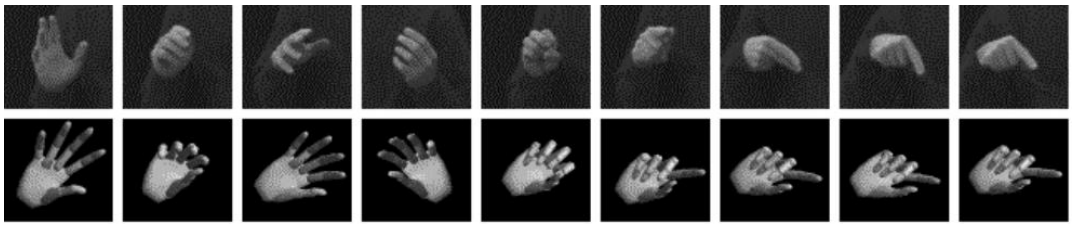

Figure 2.15: Captured Original Hand and the Nearest Predefined Corresponding Articulations [22] [23]

They capture the hand and try to map the hand's shape to a predefined template of hand shape, shown in Figures 2.15 and 2.16.[22][23].

Figure 2.16: Captured Original Hand on the Left, the Nearest Predefined Corresponding Articulation on the Right [22] [23]

2.2.2.2 Iterative calculation systems

These systems do calculations for each finger's articulation for every image. Like Inverse Kinematics (IK) algorithms they continuously calculate each articulation. Some of these IK algorithms are Jacobian Transpose, Pseudoinverse and Damped Least Squares (DLS) [24] and Cyclic Coordinate Descent (CCD) [25].

CHAPTER 3

PROPOSED METHOD

3.1 Stages

In this thesis, the resulting 3D hand articulation model reached by four distinct steps. First one is the preparation of the stereo imaging, next the capturing color markers and tracking, the third one is the conversion of the 2D points to 3D points, and the last one is the 3D hand articulations.

3.1.1 Stereo calibration

As it is used two cameras in this thesis, calibration involves stereo calibration. But first the distortion of each camera had to be handled. After that, rectification, correspondence and reprojection are addressed resulting two stereo calibrated cameras.

There are radial and tangential distortions of the cameras. These distortions are corrected using a calibration object. This object may be at any shape which can be precisely defined, like a cube or a chessboard (more like a pattern like a chessboard). In this thesis, a chessboard shape calibration object is used.

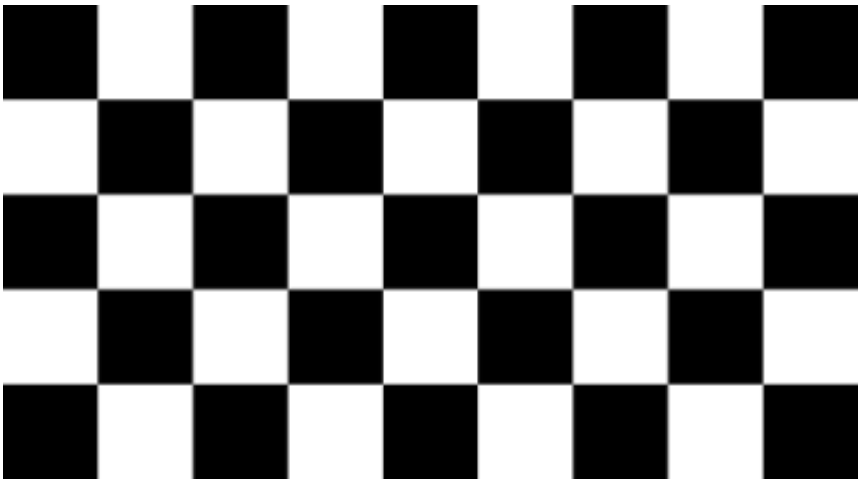

Figure 3.1: A Chessboard Shape Calibration Object

Undistortion step starts with capturing chessboard images from two cameras. These images contain the calibration object which is a chessboard image.

For each image, the corners in the chessboard pattern are found. These corners help us to solve the equations need to find the 3-by-3 homography matrix H that maps a planar object onto the imager [13].

The intrinsics, individual translations and rotations are found using these images of the chessboard. To accomplish undistortion, first the distortion map is computed. This operation is time consuming since the distortion map does not change; it is not computed every time. Using the distortion map and the intrinsics, and the images from the cameras are corrected.

Using undistorted images from two cameras, rectification is done resulting two images with the same features in each camera's image are row aligned.

The differences in the positions of the same feature in the left and right cameras are used for the calculation of the disparities. The disparity map is turned into the distance by triangulation that results to a depth map. This map will be used to calculate the 3D points of the color markers.

3.1.2 Finding – capturing color markers

In this thesis, the HSV color space chosen to eliminate the effect of amount of light. Since each color marker has a different color and the fingers are articulating, the amount of light is changing on the markers. In the HSV color space, the amount of light changes only the value (V) component. So, choosing proper hue (H) and saturation values/intervals renders capturing of the color markers easy.

Color detection became simple by removing the amount of light's effect. For each color marker, the Hue and saturation values are set by a simple tool at the beginning.

3.1.2.1 Filtering hue and saturation values – forward neighbor search

There are many pixels that fit to the hue and saturation values other than those in the color markers. Only setting the hue and saturation values are not enough. To eliminate this noise, a noise reducing algorithm, forward neighbor search is used. According to this algorithm the pixels do not have enough neighbors at their south, east and southeast coordinates are discarded

The number of the expected neighbors is optimized and a better distribution is obtained for the CAMSHIFT algorithm to work.

3.1.2.2 Finding and tracking – CAMSHIFT algorithm

Although in this thesis, CAMSHIFT algorithm is used to find the moving color markers, it is not exclusive to image processing. It can be used for any distribution changing on a dimension, like time. At every frame coming from the cameras, the color conversion done from RGB to HSV, and using the forward neighbor search noise removed, after that for each color marker, CAMSHIFT algorithm is applied; resulting the center points of each color markers as 2D points.

The search window that is given to the CAMSHIFT at the beginning is the whole frame. After this first frame, the center point of each color marker is found, and a rectangle that contains these points drawn. And another rectangle

that is larger than the first rectangle is calculated with a factor,this factor is calculated considering the inner rectangle's width and height.

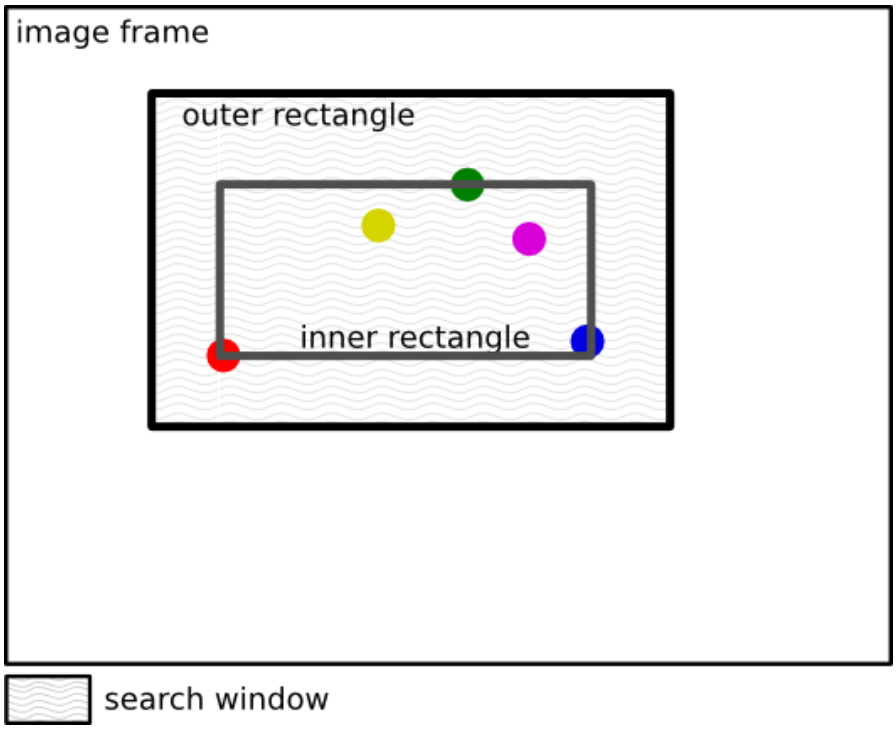

Figure 3.2: Five Color Markers, Inner and Other Rectangles and Search Window

The outer rectangle is used as the search window of the color markers in the next frame, and it is defined in every frame. The inner and the outer rectangles and the search window are shown in the Figure 3.2.

3.1.3 Conversion of marker's 2D point to 3D point

The conversion from 2D points to 3D points accomplished as it is described in the reprojection step of the stereo imaging. On the rectified images, same color marker's center point found as a 2D point in left and right image. The 3D points are calculated using the equations in the reprojection step of the stereo imaging. These calculations done in each frame acquired from cameras, therefore in every frame the center points of the color markers are found as 3D points.

3.1.4 Calculation of the articulations, geometric calculation and IK

There are assumptions regarding the articulation of the thumb, therefore the calculation of this articulation is easier and it is done in one step without iterations, so it is faster. The articulations of other four fingers are calculated according to the equation that gives the relation of the angles and it is iterative.

3.1.4.1 Index to little fingers articulation

The fingers from Index to Little have same amount of phalanxes and joints, and they have similar articulations.

3.1.4.1.1 Iterative calculations and relationship of four fingers' angles

These four fingers are assumed to articulate in two dimensions, on the y and the z axis and it is assumed that each of three phalanxes at equal length. For each frame, the 3D point of the color marker that is attached to the fingertip is calculated. From this 3D point, only the y and z components are used and the articulation of the finger calculated with these values, the articulation on the y-z plane is shown in the Figure 3.3.

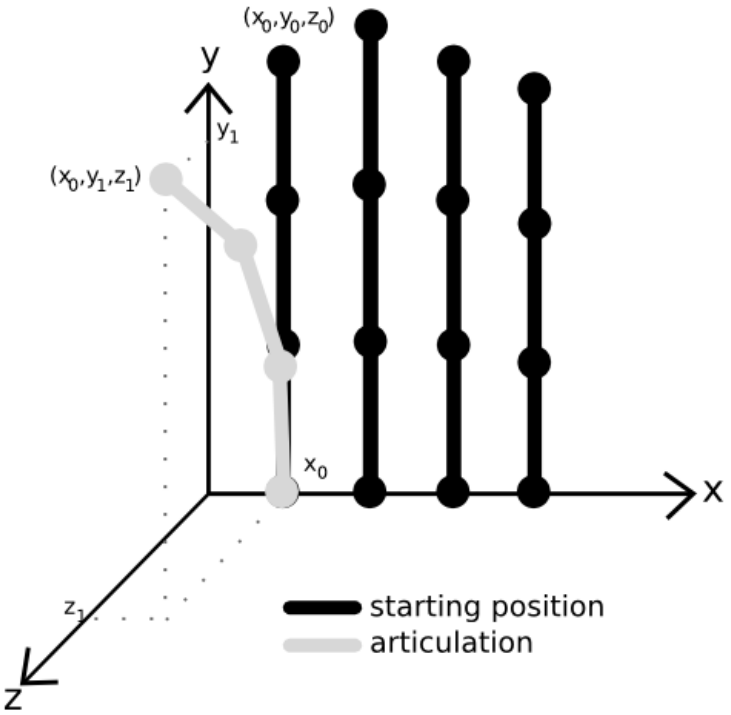

Figure 3.3: The Articulation in the y-z Plane

The starting position of each finger is the maximum extend of that finger. Along the articulations the x value is always assumed as the starting x value x_0.All the calculations are done in the y-z plane. The calculation of the articulation of the finger related with the below equation;

$$q_{\alpha,\beta} = 0.23 + 1.73d + 1.5d^2 \quad (3.1)$$

Here α and β angles are the outer angle and the middle angle of the finger, respectively. The first angle which is between the first phalanx and the y-axis is represented with θ_1, these angles are shown in the Figure 3.4. And the d is the ratio of the distance from origin to fingertip over maximum distance.

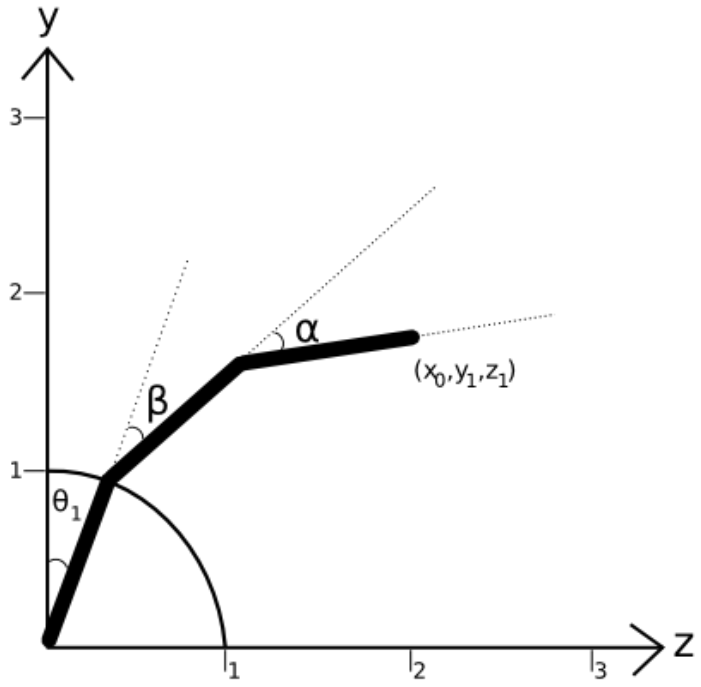

Figure 3.4: The Angles α, β and the θ_1

For a target point (T) as (x_0, y_1, z_1) the articulation is calculated according the position of the T. A number of calculation steps are involved, all the points, angles and the lines are shown in the Figures 3.5 and 3.6. These steps are summarized like below;

i. Draw a line from origin to target point, line *t*.

ii. Find the angle between this line and the y-axis, angle θ. The angle ofθ_1 is the half of θ angle.

iii. Draw a line with an angle of θ_1 from origin, with length of 1 unit to the point P_1.

iv. Draw a line from P_1 to target point T, line *l*.

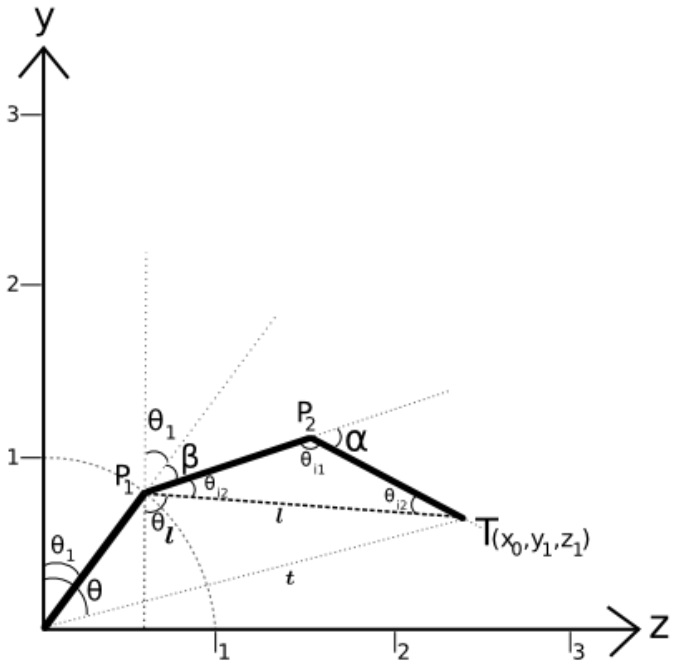

Figure 3.5: The Angles and the Points of the Articulation of One of the Four Fingers, *l* with a Decreasing Slope.

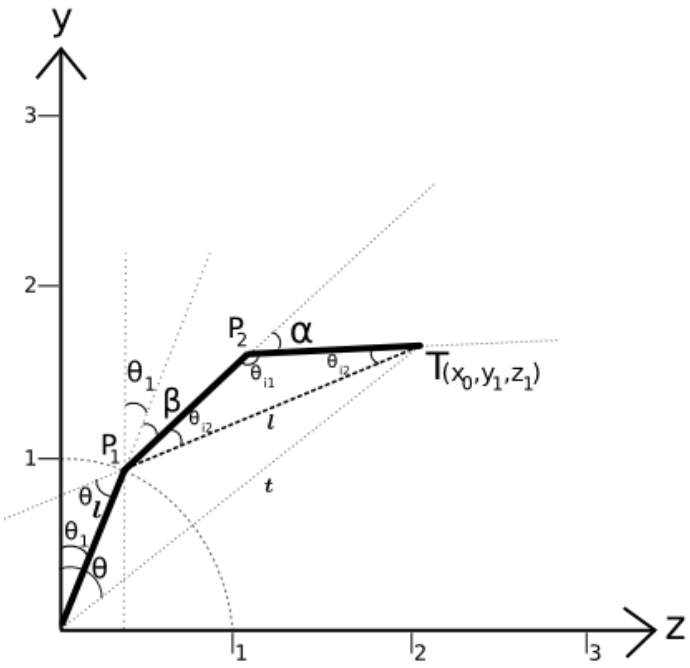

Figure 3.6: The Angles and the Points of the Articulation of One of the Four Fingers, *l* with an Increasing Slope.

v. Calculate the α angle,

There are two other phalanxes need to be considered for the articulation. One of them positioned from the P_1 to P_2, and the other one is positioned from P_2 to T.

Since all three phalanx of the finger assumed equal length, these two phalanxes and the *l* create an equilateral triangle. The angles of this rectangle are θ_{i1} and two θ_{i2}.

The θ_{i1} is calculated using the cosine theorem,

$$c^2 = a^2 + b^2 - 2ab\cos C \tag{3.2}$$

Since the *l* is a know length from P_1 to T, and the unit of one phalanx is 1 unit, the θ_{i1} can be calculated,

$$l^2 = 1^2 + 1^2 - 2.1.1\cos\theta_{i1} \tag{3.3}$$

Therefore the θ_{i2},

$$\theta_{i2} = \frac{\pi - \theta_{i1}}{2} \quad (3.4)$$

Angle α is calculated as,

$$\alpha = \pi - \theta_{i1} \quad (3.5)$$

vi. According the slope of the *l*, if it is decreasing or not, the angle β is calculated,

If the slope is decreasing,

$$\beta = \pi - (\theta_1 + \theta_{i2} + \theta_l) \quad (3.6)$$

Otherwise,

$$\beta = \theta_l - (\theta_1 + \theta_{i2}) \quad (3.7)$$

vii. The θ_l is calculated using the half PI minus the arc tangent of the slope of line *t*.

$$\theta_l = \frac{\pi}{2} - \text{atan}(slope(t)) \quad (3.8)$$

viii. Each phalanx's length is 1 unit, so the maximum length of the finger is 3 times of that, and the d value in calculated like,

$$d = \frac{t}{3*1} \quad (3.9)$$

ix. Now α, β angles and the *d* value are found, using the equation the $q'_{\alpha,\beta}$ value calculated.

x. Check the $q'_{\alpha,\beta}$ against standard $q_{\alpha,\beta}$ value.

a. If the $q'_{\alpha,\beta}$ is greater than the standard value of $q_{\alpha,\beta}$, it means that the *l* must be greater than it is now, and the θ_1 must be less than it is now, so set the$\theta_1$to its half value, then go back to step iii.

If ($q'_{\alpha,\beta} < q'_{\alpha,\beta}$) then

$\theta_1 = \theta_1 / 2$

go to step_iii

b. If the $q'_{\alpha,\beta}$ is less than the standard value of $q_{\alpha,\beta}$, it means that the *l* must be less than it is now, and the θ_1 must be greater than it is now, so increase the value of $\theta_1$with half of it, then go back to step iii.

If ($q'_{\alpha,\beta} > q'_{\alpha,\beta}$) then

$\theta_1 = \theta_1 + (\theta_1 / 2)$

go to step_iii

c. If the $q'_{\alpha,\beta}$ is equal to the standard value of $q_{\alpha,\beta}$, then continue.

In this thesis the difference between these two values, considered with a value of difference tolerance. If the difference is less than or equal to this tolerance value than it is assumed equal. The effect of difference tolerance is explained in detail under the "The Effect of Difference Tolerance".

d. If the $q'_{\alpha,\beta}$ is never equal to the standard value of $q_{\alpha,\beta}$ (or the difference never less than the tolerance)this may end with an infinite loop, to prevent that this loop is limited with a maximum loop count. When reached to that maximum loop count, the values of the closest $q'_{\alpha,\beta}$ is accepted as the solution.

xi. It is now found that all the necessary values; θ_1, α and β angles to define and draw the articulation.

Using the data calculated until this point the 3D points at the origin, P_1, P_2 and T can be calculated and drawn in a 3D model. These points' x-axis values are set as the x_0 value, since it is assumed that the finger cannot move along the x-axis.

3.1.4.1.2. The effect of difference tolerance

The computed $q'_{\alpha,\beta}$ value may not be equal to the standard $q_{\alpha,\ \beta}$ value. An iterative approach may let the $q'_{\alpha,\beta}$ value to approximate to the $q_{\alpha,\ \beta}$ value. But most of the time this process is time consuming and CPU intensive. To eliminate this undesired effect, $q'_{\alpha,\beta}$ value is not expected to be equal to $q_{\alpha,\ \beta}$ value, but it should be close. The accepted value of difference between the $q'_{\alpha,\beta}$ value and the $q_{\alpha,\ \beta}$ value is defined as "Difference Tolerance" (DifT).

DifT value is an optimization between time (CPU usage) and accuracy. Although the target point and the origin will not change, the points P_1 and the P_2 values may slightly change.

3.1.4.1.3. Solution space

A finger's articulation is bound by minimum and maximum equations. If a point is inside this solution space the articulation calculated according to that point, this solution space is shown in the Figure 3.7.

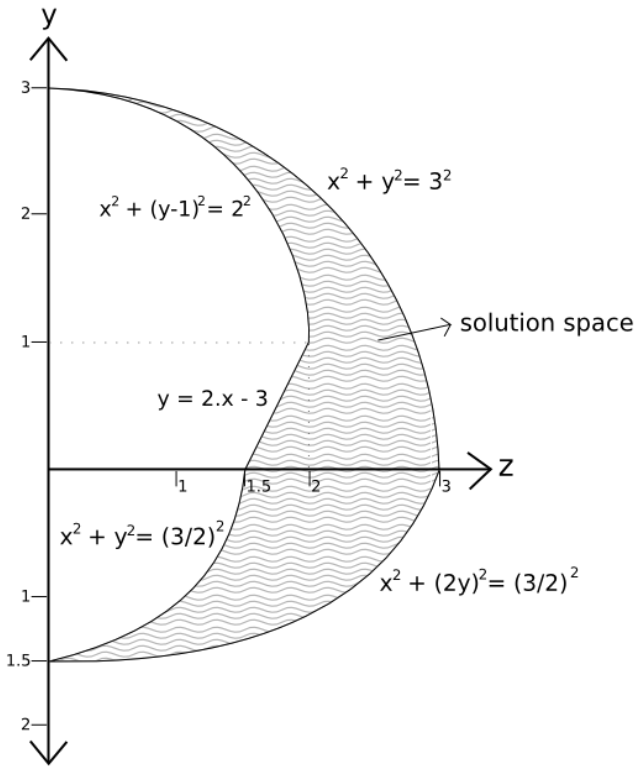

Figure 3.7: The Solution Space of the Four Fingers

Those points outside of the solution space calculated accordingly,

- If a point at the right side of the solution space and,
    - The y value is positive than its y value's projection found on the $x^2+y^2=3^2$
    - The y value is negative than its y value's projection found on the $x^2 + (2.y)^2= (3/2)^2$

- If a point at the left side of the solution space and,
    - The y value is positive and greater than 1 unit, than its y value's projection found on the $x^2 + (y-1)^2 = 2^2$
    - The y value is positive and less than 1 unit, than its y value's projection found on the $y = 2.x - 3$
    - The y value is negative, than its y value's projection found on the $x^2 + y^2 = (3/2)^2$

Although the human finger may do articulations resulting the target point outside this solution space, this solution space and its boundaries are assumed for a faster result and less CPU intensive solution.

3.1.4.2. Calculation of thumb articulation

The thumb's articulations are different from other four fingers, as a result thumb's articulations are handled separately.

3.1.4.2.1 Geometric solution

The thumb's articulations are complex and some of the target points may have more than articulation one solution. Trying to find them all, or investigating the possibility of other solutions for a point may result to long response time and extensive CPU usage, to eliminate this, only the outer most solution of the thumb articulation is addressed and all three phalanxes of the thumb are assumed at equal length, this is show in the Figure 3.8.

For a known target point T (x_T, y_T, z_T), the points P_1 and P_2 has to be found to draw the articulation of the thumb. To find the points P_1 and P_2 there are many

calculation steps involved like finding the value of *d*, the distance from origin to target point, or the l_T, the projection of the target point's on the z-x plane.

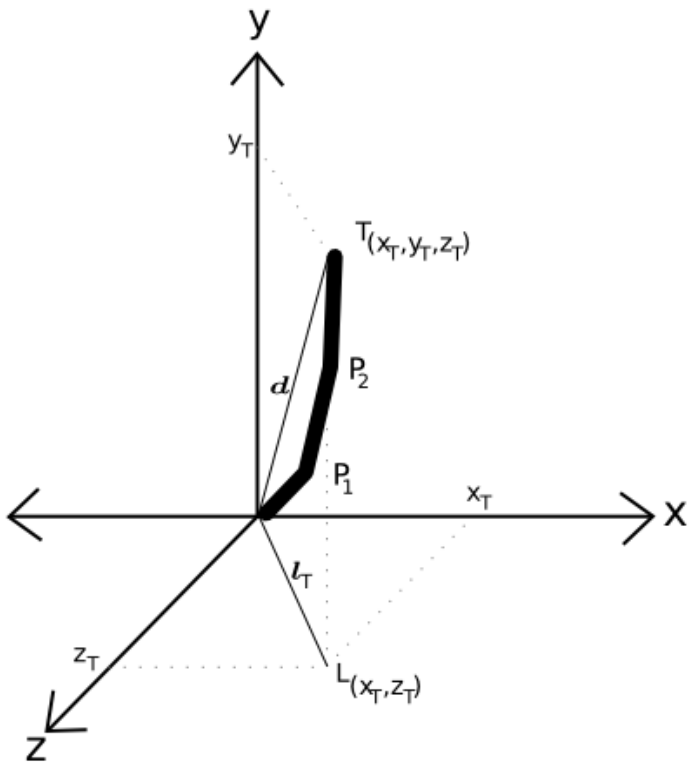

Figure 3.8: The Outer Most Solution of the Thumb Articulation

The point L (x_T, z_T) is the projection of the target point T (x_T, y_T, z_T) on the z-x plane, this is shown in the Figure 3.9.The calculations of the distance l_T and the angle θ_T are follows,

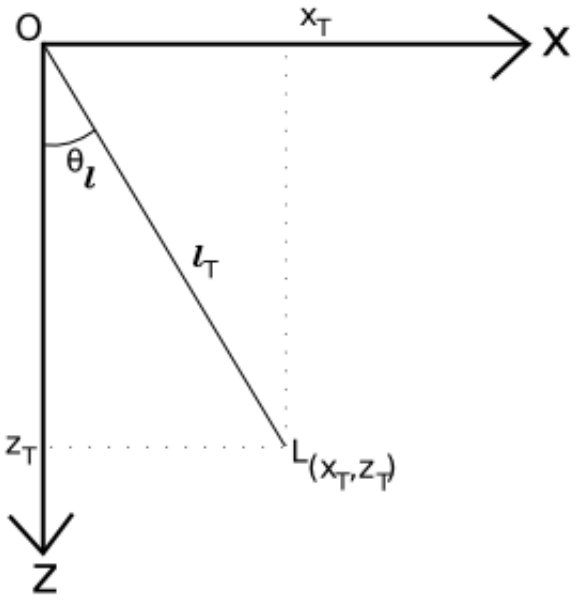

Figure 3.9: The Projection of the Target Point on z-x Plane

The l_T is calculated from x_T and z_T values,

$$l_T=\sqrt{{x_T}^2+{z_T}^2} \tag{3.10}$$

The θ_T found from the arc-tangent of the x_T over z_T

$$\theta_{l_T=}atan\left(\frac{x_T}{z_T}\right) \tag{3.11}$$

The distance between the origin and the target point d calculated,

$$d_T=\sqrt{{y_T}^2+{l_T}^2}=\sqrt{{y_T}^2+{x_T}^2+{z_T}^2} \tag{3.12}$$

Below graphic describes the thumb's articulation on the y-l plane, Figure 3.10,

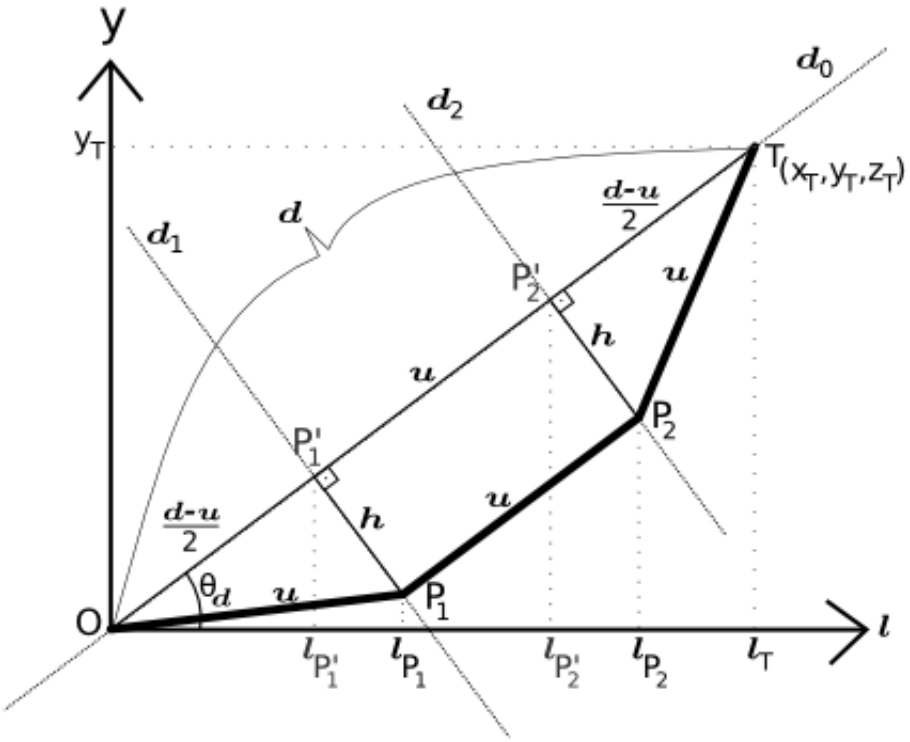

Figure 3.10: Calculation of the Thumb's Articulation

Here the d_0 line is the line passing through origin and the target point T. The θ_d is the angle between the d_0 line and the *l* axis.

The angle θ_d is calculated from the arc-tangent of the y_T over l_T

$$\theta_{d=} atan\left(\frac{y_T}{l_T}\right) \tag{3.13}$$

Each three phalanxes are equal length and it is represented by *u*. In order to calculate the position of the P_1, first the P'_1 has to be calculated. The point P'_1 is at a point on the line d_0, at a distance of (*d*-*u*)/2. The point P'_1 in the y-*l* plane is defined as y and *l* points, these can be calculated using θ_d and the distance (*d*-*u*)/2.

$$P'_1 \Longrightarrow \theta_d\ , \left(\frac{d-u}{2}\right) \Longrightarrow \left(y_{P'_1}, l_{P'_1}\right) \tag{3.14}$$

$$y_{P'_1}{=}sin(\theta_d)\cdot\left(\frac{d-u}{2}\right) \tag{3.15}$$

$$l_{P'_1}{=}cos(\theta_d)\cdot\left(\frac{d-u}{2}\right) \tag{3.16}$$

The x and z values of the P'_1 can be calculated as,

$$x_{P_1'}=sin(\theta_{l_T})\cdot l_{P_1'} \tag{3.17}$$

$$z_{P_1'}=cos(\theta_{l_T})\cdot l_{P_1'} \tag{3.18}$$

These calculations also done for the point P'_2 and all the x, y and z values of this point calculated.

In the Origin (O), P'_1, P_1right triangle, two of the edges are known as u and the (d-u)/2, therefore the third edge h, can be calculated,

$$h=\sqrt{u^2-\left(\frac{d-u}{2}\right)^2} \tag{3.19}$$

To find the line d_1, d_0 is rotated about half of the π clockwise, around the point P'_1. On the lined_1, at a distance of h, the point P_1 can be found with the values of y and l.

The point P_1 has the y and l values, and the x and the z values can be calculated as,

$$x_{P_1}=sin(\theta_{l_T})\cdot l_{P_1} \tag{3.20}$$

$$z_{P_1}=cos(\theta_{l_T})\cdot l_{P_1} \tag{3.21}$$

The line d_2 also found as the same method used for the line d_1, and the point P_2 is calculated as the same way the point P_1 calculated.

At this point the articulation of the thumb can be described with these points; the origin (O), points P_1 and P_2, and the target point T.

3.1.4.2.2 Solution space

For the thumb articulation, the solution space is shown in the Figure 3.11.

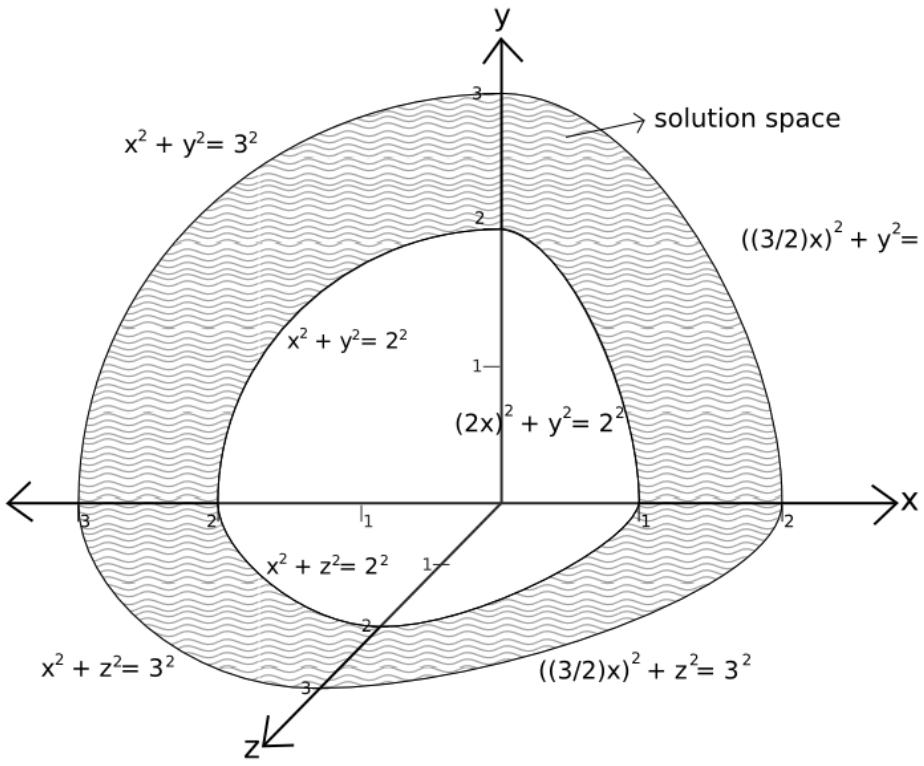

Figure 3.11: The Solution Space of the Thumb's Articulation

The points inside the solution space are used for the calculations of the thumb's articulations. The points that are not in the solution space are projected on the closest curve, and the thumb articulation calculated according to that projected point.

3.1.5. 3D hand reconstruction

All the joint points for five fingers are calculated and using the points a 3D hand model can be reconstructed. The related points of each finger are represented as 3D points in a 3D space and lines drawn between these 3D points to form the 3D fingers. Since the starting positions of fingers known, a 3D hand model is constructed in this 3D space.

3.2 Classify – Compare

There are different approaches to model a 3D hand or to model an interface to capture the articulations of a hand and there are hardware and software aspects of them.

3.2.1 Hardware systems

Although the hardware systems are accurate, they involve various devices and tools like transmitters, detectors and gloves [8] [20] [21] that may not be found at any time anywhere or they may be expensive. Not just the tools and devices but they may also involve special training. The wearable hardware based methods like gloves, digitize the finger articulations into multi-parametric data. The sensors attached to the glove make it easy to collect articulations and movements. The devices may be quite expensive and they may not be accepted easy by users since they present cumbersome user experience [26].

3.2.2 Software systems

These systems do not use special tools and devices, they acquire images from cameras and the images are processed in accordance with the corresponding technique. These systems can be explained under learning, training, fitting systems and iterative calculation systems.

The first group is adaptive but those involve training may need time for training, and the fitting systems need to do calculations for each position of the hand to map a predefined hand posture [22] [23], and this may lead to a different but close representation of the hand, and also this process is CPU intensive. For rough matching to an already defined hand posture may be calculated in very short time, but especially if a precise matching is needed this time rises exponentially to a value that is not acceptable [27]. In this group there are algorithms and approaches are used like Hidden Markov Models (HMM), particle filtering and condensation, finite-state machine (FSM), skin color segmentation, neural networks [28].

The second group does the calculations every time, and according to the implemented algorithm like Inverse Kinematics (IK) algorithms [24] [25] they may end up with impossible or wrong articulation and the correct calculations may be CPU intensive.

3.2.3 Proposed method

The proposed method in this thesis falls into the group of software systems, with usage of pre-calculated functions, geometric calculations. It is different from the pure inverse kinematics solutions in this sense.

3.2.4 Pros and cons

The proposed method in this thesis has advantages and disadvantages concerning the accuracy, availability, cost and intuitiveness.

3.2.4.1 Accuracy

Hardware systems designed to be precise and accurate. Although the proposed method does not accurate as the hardware systems because the aim is not that, but the proposed method is not limited with a number of predefined hand articulations and it can model all the articulation of a hand within the assumptions.

3.2.4.2 Availability

The proposed method is readily available to any application with a computer that can run the software with two cameras. On the other hand the hardware systems may not be acquired or they may not be applicable for every application.

3.2.4.3 Cost

The proposed method is affordable while it only needs two web cameras and a commodity computer. Since the computation is not intense the computer may be a portable computer, a laptop. The fitting systems may need more CPU power, memory and disk space to store the predefined hand postures, and also the hardware systems as in their nature; they have a cost of tools and devices.

3.2.4.4 Intuitiveness

Intuitiveness of an interface may be defined as the familiarity of the interface [3]. From this definition it is clear that the hand of a human can be interpreted as an intuitive interface for computer interactions. The proposed method models the articulations of the hand within the assumptions, for this modeling to be accomplished only the color markers at the top of the fingers are involved. Leaving the color markers aside, it can be said that this method is an intuitive interface for computer interaction.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Accomplished

In this thesis, these steps are accomplished; the stereo calibration, color markers detection, tracking of the color markers, finding 2D points and conversion to 3D points, finding the articulations of the fingers, modeling a 3D hand that shows the articulations.

4.1.1 Stereo calibration

In the stereo calibration step, multiple chessboard images are captured from two cameras at the same frame to accomplish the undistortion. These images are shown in the Figure 4.1.

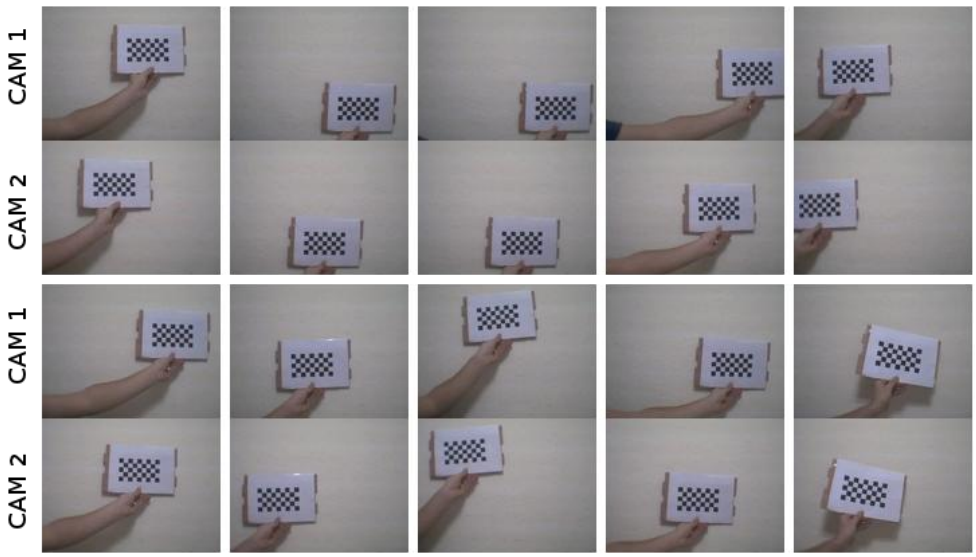

Figure 4.1: Captured Chessboard Images at the Same Frame from Two Cameras

For each captured image, the corners are found; the corners are shown in the Figure 4.2.

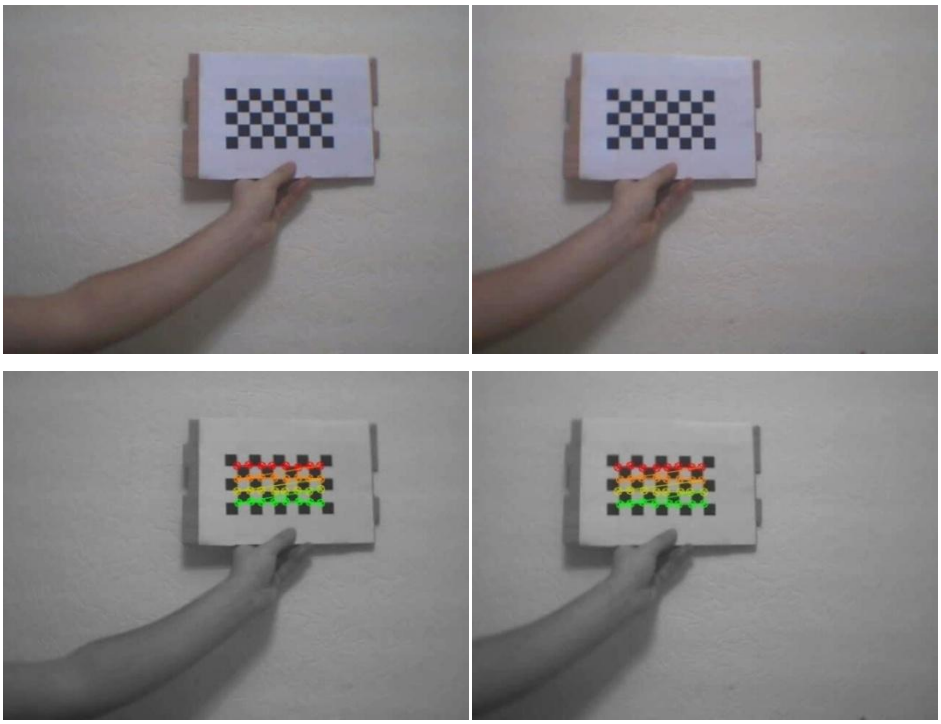

Figure 4.2: Top Left Original Image from Left Cam, Top Right Original Image from Right Cam, Bottom Left and Right are The Corresponding Images that the Corners of the Captured Chessboard Images are Found.

The necessary calculations done and the cameras are corrected using the distortion map and the intrinsics. Rectification is done with these undistorted images; resulting row aligned images, the rectified images are shown in the Figure 4.3.

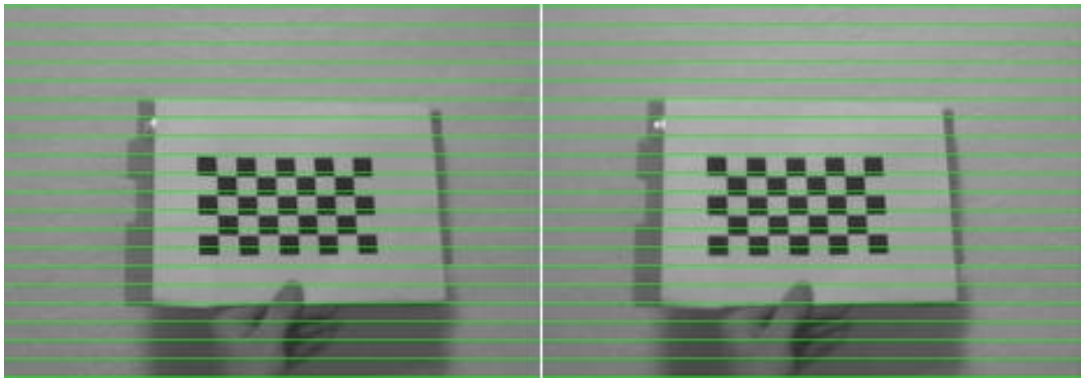

Figure 4.3: Left and Right Camera Images After Rectification

After rectification, the disparity map is turned into the distance to result a depth map by triangulation. 3D points of the color markers' are calculated from this map.

4.1.2 Color detection

Since in this thesis, the color space is chosen as HSV color space, and the V (value) component omitted. Setting the correct values/intervals for hue (H) and saturation (S) make it easy to capture color markers. Using a simple tool developed for this thesis, the hue and saturation values are set.

4.1.2.1 Setting hue values

The hue value of each color marker is represented by a 16 bar histogram. This histogram's top most two values are selected and the rest of the values are discarded, this histogram is shown in the Figure 4.4.

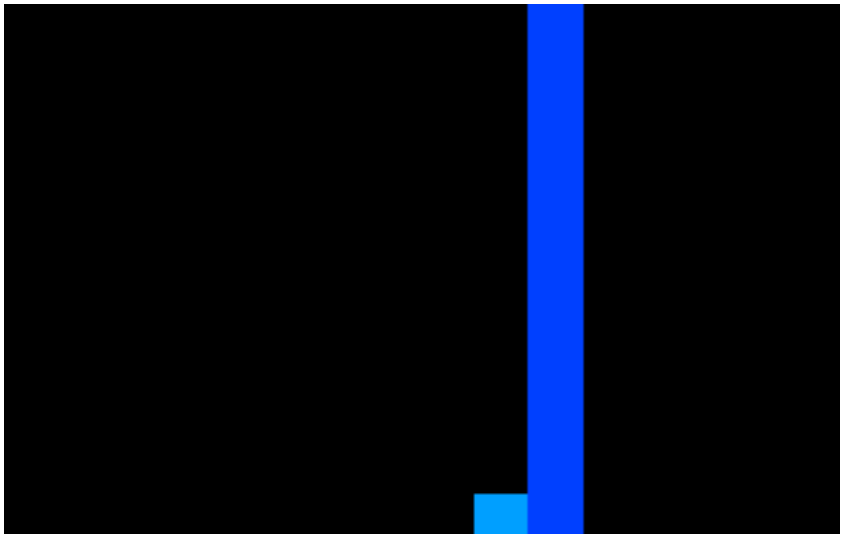

Figure 4.4: The Histogram of a Color Marker's Hue Value

4.1.2.2 Setting saturation values

For the saturation values, a range is selected for each color marker. The selection of the correct range of the saturation is important otherwise there will be excess amount of the noise, the noise and the selection of correct range shown in the Figure 4.5.

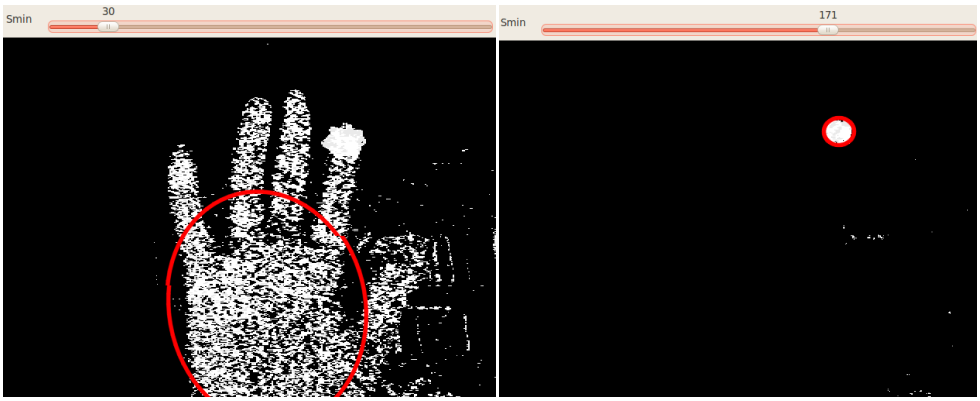

Figure 4.5: The Wrong Saturation Range and the Noise on the Left, the Correct Range with Minimum or No Noise on the Right

4.1.3 Tracking the color markers

Tracking of the color markers done using the CAMSHIFT algorithm but to get better and accurate results from CAMSHIFT, the forward neighbor search algorithm used to remove the noise. The forward neighbor search algorithm reduced the size of the search window for the CAMSHIFT, this lead to a faster result. The removal of the noise using forward neighbor search is shown in the Figure 4.6.

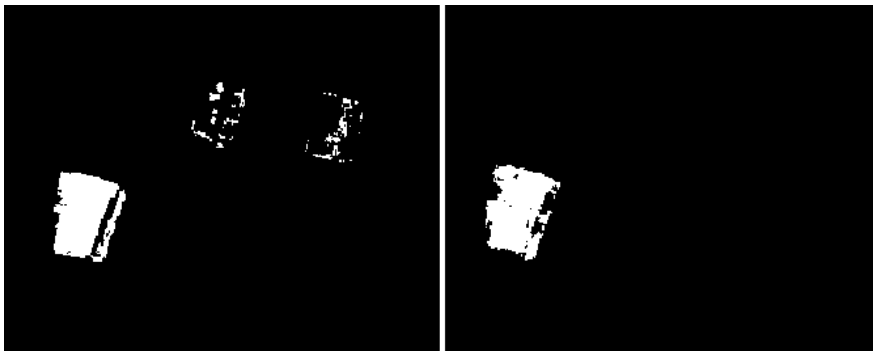

Figure 4.6: Image with Noise on the Left, Image without Noise on the Right, Result of Forward Neighbor Search

The noise is removed from images and now the search window size is smaller for CAMSHIFT to result better. In the Figure 4.7 on the left, it is shown that the forward neighbor search removes almost all the noise, and on the right the search windows are shown for each color marker. Except for the pink colored marker at the ring finger, all the other four fingers' search windows' sizes are very small, almost to the dimensions of each corresponding marker; this is the

result of the noise removal. But for the pink colored marker at the ring finger, the search window is the largest. This is because there are some pixels in the red colored marker at the thumb that they fit in the values/ranges of the hue and saturation of the pink colored marker. These pixels could not be removed by the forward neighbor search, and the result is a rather large search window.

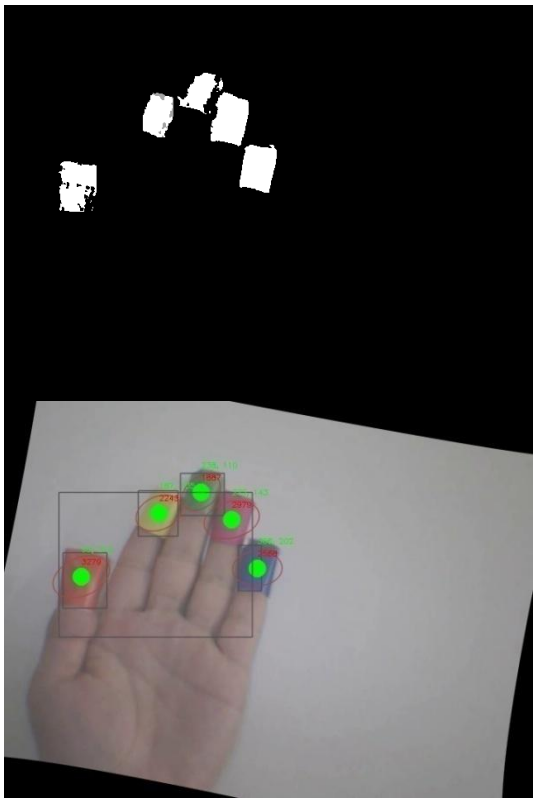

Figure 4.7: On the Left, Noise Removed Image, on the Right the Search Windows for Each Color Marker.

For each captured frame searching the whole frame for corresponding hue and saturation values and doing noise removal to full size frame is CPU intensive and time consuming. After all color markers are found, not the whole frame but a fragment of the image is investigated. This fragment's size and position calculated as;

- First the minimum and maximum values of x and y are found from the set of the color markers' center points

- After that, the inner rectangle is drawn from minimum point P_{min} to maximum point P_{max}
- The outer rectangle is drawn outside of the inner rectangle with the dimensions from 0 to 50 percent greater than the inner rectangle at the points P'_{min} and P'_{max}. The dimensions of the outer rectangle increases with the speed of the articulations. Also the dimensions increase if the starting position of the hand gets closer to the cameras.

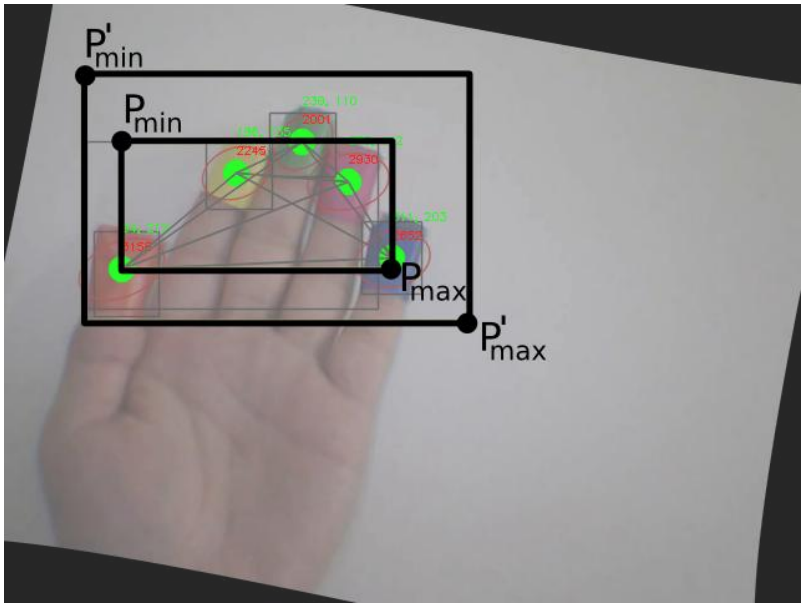

Figure 4.8: The Inner and the Outer Rectangles

4.1.4 Articulations

The articulations are considered under two categories, first one is the thumb articulations and the second one the other four fingers' articulations

4.1.4.1 Four fingers articulations

Since it is assumed that these fingers will not be able move in the x-axis, a 2D tool developed to show the articulations of these fingers in y-z plane. Using this tool the articulation of a finger is demonstrated, and the joints and phalanx can be seen, as shown in the Figure 4.9

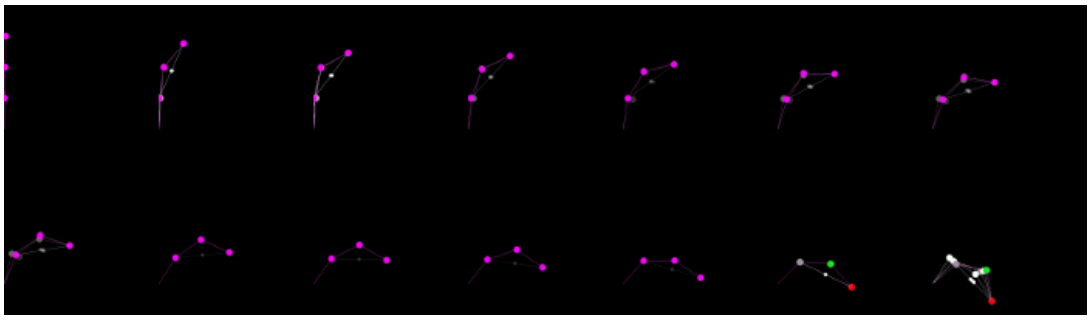

Figure 4.9: Articulation of One of the Four Fingers from Index to Little Finger

In the Figure 4.9 the last two articulations has a different color. The articulation of the finger is calculated in an iterative approach. The joint angles are computed in every iteration and than they are checked against the equation below.

$$q_{\alpha,\beta} = 0.23 + 1.73d + 1.5d^2 \qquad (4.1)$$

The angles α and β are the middle and the outer angles of the finger respectively, and the d is the distance between the base joint and the fingertip relative to the finger length.

If the computed and the standard values are equal or the difference between these values is less than the difference tolerance than this articulation assumed correct. But if the maximum iteration count reached than the closest values of the joint angles are assumed as correct articulation, this is shown in the Figure 4.10.

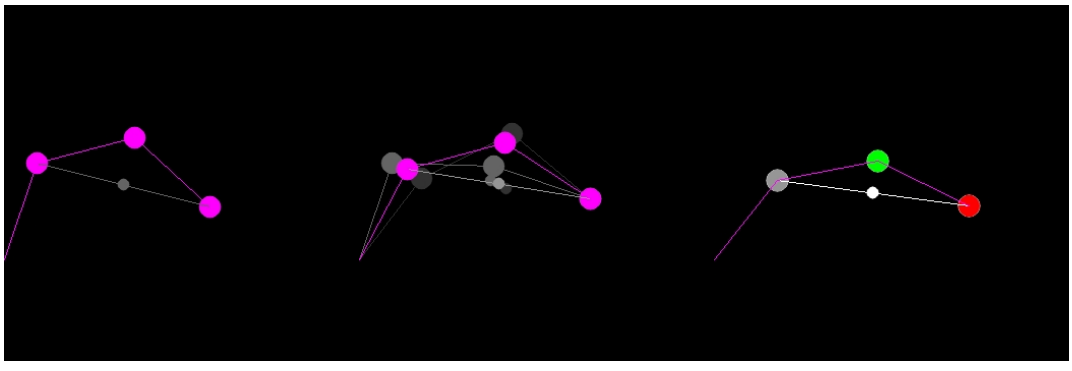

Figure 4.10: On the Left; Solution Found at First Iteration, at the Middle; Solution Found at Third Iteration, on the Right; No Solution within Maximum Iteration Count; for that Chose the Closest Solution

The DifT value is optimized to use less CPU with faster response time. For the same target point, three DifT values and their corresponding results investigated.

Table 1: Iterative Search for Difference Tolerance Values Table, with the Angles of α, β, θ1

Loop count	difference	α	β	θ_1
1	1.348579	0.757710	0.234905	0.200871
2	1.877152	0.137273	0.822010	0.301307
3	**0.770254**	**0.561584**	**0.474559**	**0.451960**
4	4.711385	0.828657	0.126401	0.225980
5	1.579100	0.309925	0.702461	0.338970
6	**0.223174**	**0.645983**	**0.379985**	**0.508455**
7	0.689847	0.724595	0.280717	0.190671
8	2.693019	0.797957	0.175305	0.214504
9	1.701419	0.246986	0.749164	0.321757
10	11.994678	0.865616	0.062630	0.241317
11	1.424376	0.377172	0.648346	0.361976
12	0.216811	0.690465	0.325321	0.180988
13	1.701419	0.246986	0.749164	0.321757
14	0.493773	0.609428	0.422308	0.482635
15	11.994678	0.865616	0.062630	0.241317
16	1.424376	0.377172	0.648346	0.361976
17	0.216811	0.690465	0.325321	0.180988
18	1.114836	0.480499	0.555945	0.407223
19	1.565745	0.766152	0.222783	0.203612
20	1.835851	0.165416	0.804332	0.305417
21	0.717871	0.571635	0.463860	0.458126
22	5.532709	0.836447	0.113442	0.229063
23	1.547582	0.324656	0.690988	0.343594
24	**0.142914**	**0.655302**	**0.368834**	**0.343594**

The first DifT value is 0.2, and it has reached to this DifT at 24^{th} loop. The points P_1 and P_2 of the resulting articulation found as;P_1 (33.687365, 94.154986), P_2 (99.146823, 169.675713). This is shown in the Figure 4.11.

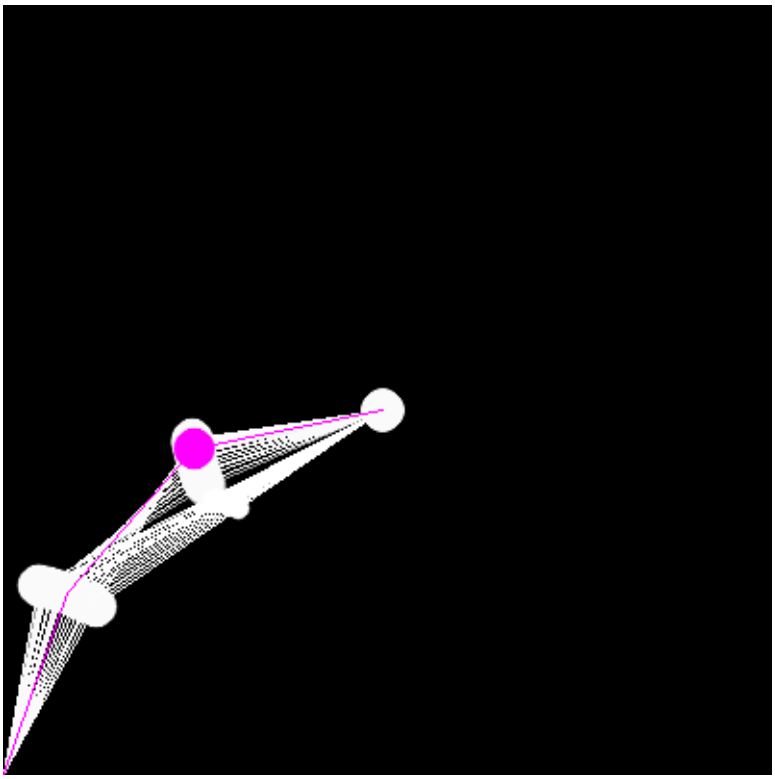

Figure 4.11: The Iterations of the Solution with a DifT Value of 0.2

The second DifT value is 0.5, and it has reached to this DifT at 6^{th} loop. The points P_1 and P_2 of the resulting articulation found as; P_1 (33.251572, 94.309771), P_2 (99.485019, 168.919908). This is shown in the Figure 4.12.

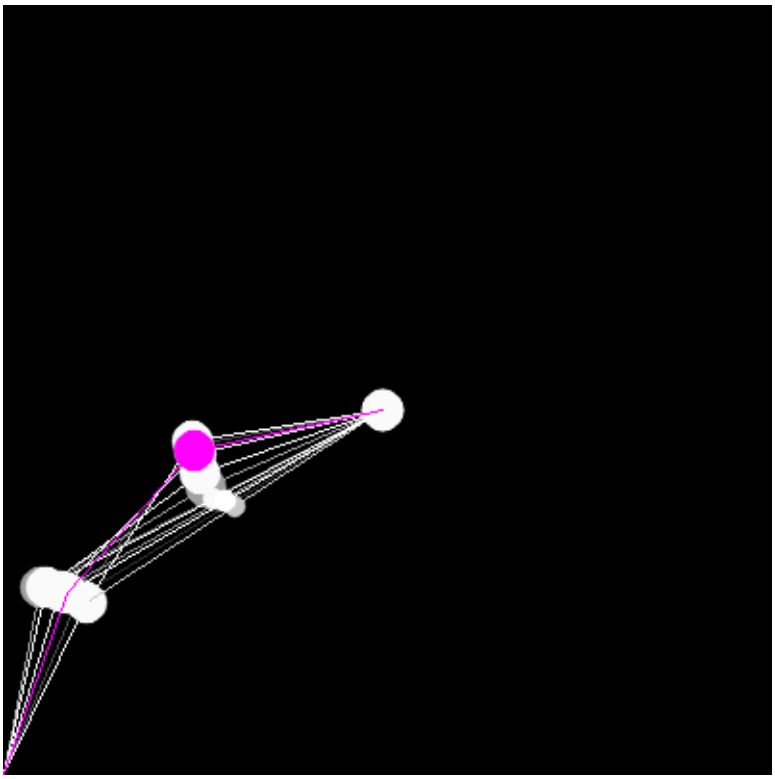

Figure 4.12: The Iterations of the Solution with a DifT Value of 0.5

The third DifT value is 0.8, and it has reached to this DifT at 3rd loop. The points P_1 and P_2 of the resulting articulation found as; P_1 (29.676812, 95.494957), P_2 (100.060201, 166.256823). This is shown in the Figure 4.13.

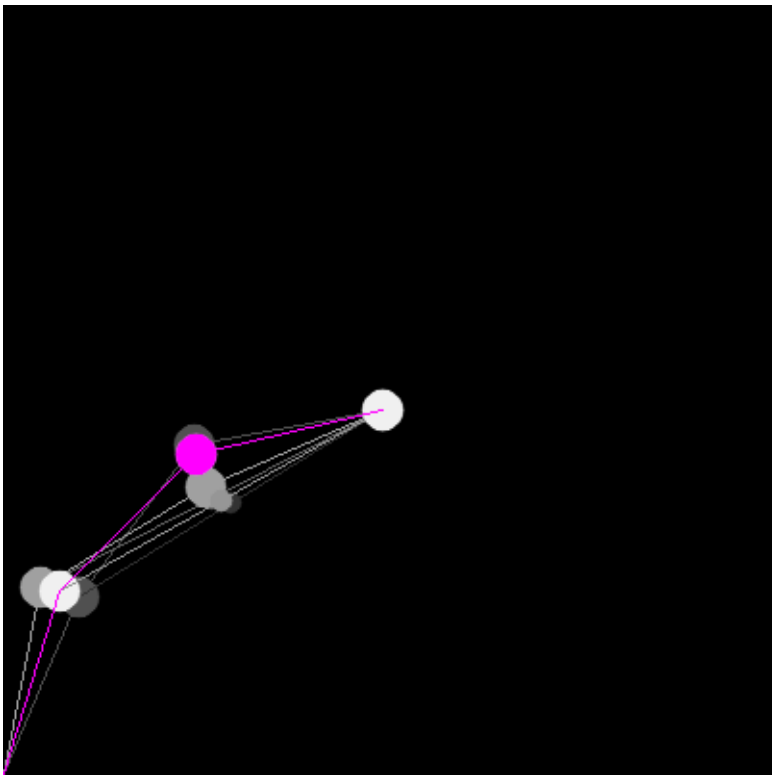

Figure 4.13: The Iterations of the Solution with a DifT Value of 0.8

As it can be seen, the DifT value of 0.5 has almost the same P1 and P2 values with the DifT 0.2 value, but it has reached to a solution within an acceptable number of steps. The DifT value optimization is done using this method.

4.1.5 3D hand reconstruction

All the 3D points of the fingers are calculated and these points connected with lines relevant lines to form a 3D hand, and this 3D hand drawn at every captured frame. Result is the reconstruction of the same 3D hand articulation of the real human hand's articulation. The articulations of the four fingers, from Index to Little finger, are shown in the Figure 4.14.

Below the articulations of the four fingers can be seen.

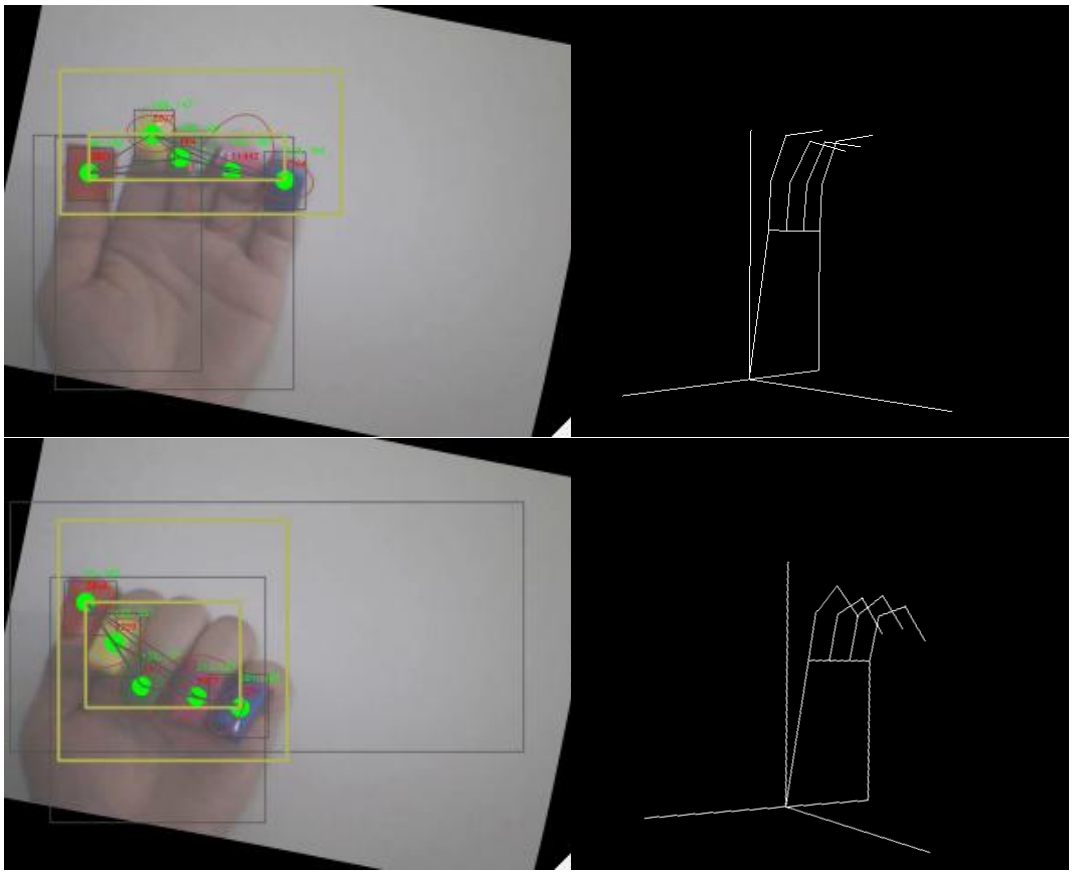

Figure 4.14: The 3D Reconstruction of the Articulations of the Four Fingers, Index to Little Fingers

And the 3D articulation of the thumb also reconstructed since the thumb articulations are calculated using the geometric method, this articulation is shown in the Figure 4.15.

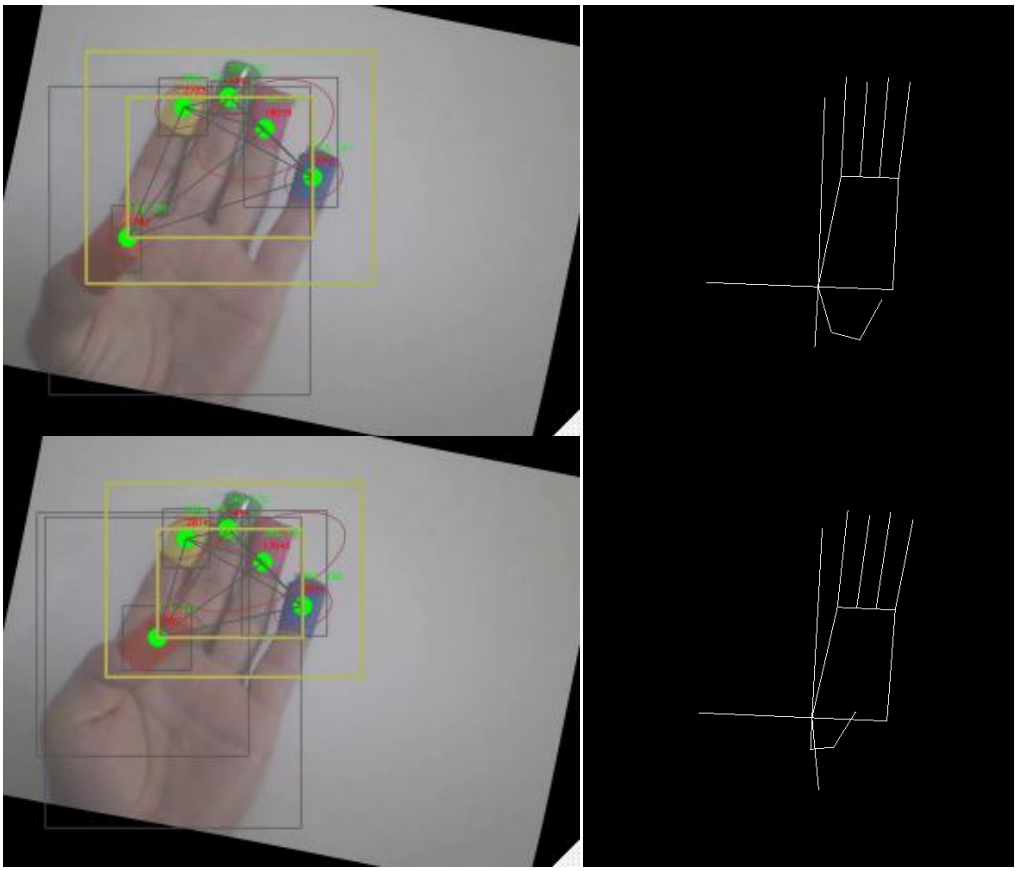

Figure 4.15: The 3D Articulation of the Thumb

4.1.6 Effect of self-occlusion

There are some articulations that one finger obstructs the visibility of another finger(s). It is assumed that the fingers from index to little finger are not capable of moving along the x-axis which in practice they do. Most of the time the articulations of these four fingers are not in the x-axis or their movement in this axis is negligible; their main movement is on the y-z plane.

But there are cases that these fingers move in front of one and other, this is shown in the Figure 4.16 and Figure 4.17. Since the movement in the x-axis is not taken under consideration, the articulations of these four fingers modeled according to the y and z values.

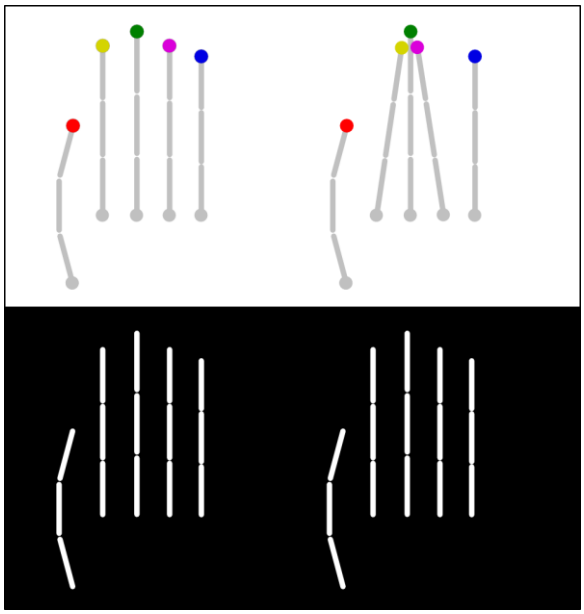

Figure 4.16: At the Top the Movement of the Index and Ring Fingers Along the x-axis, at the Bottom the Modeled Hand.

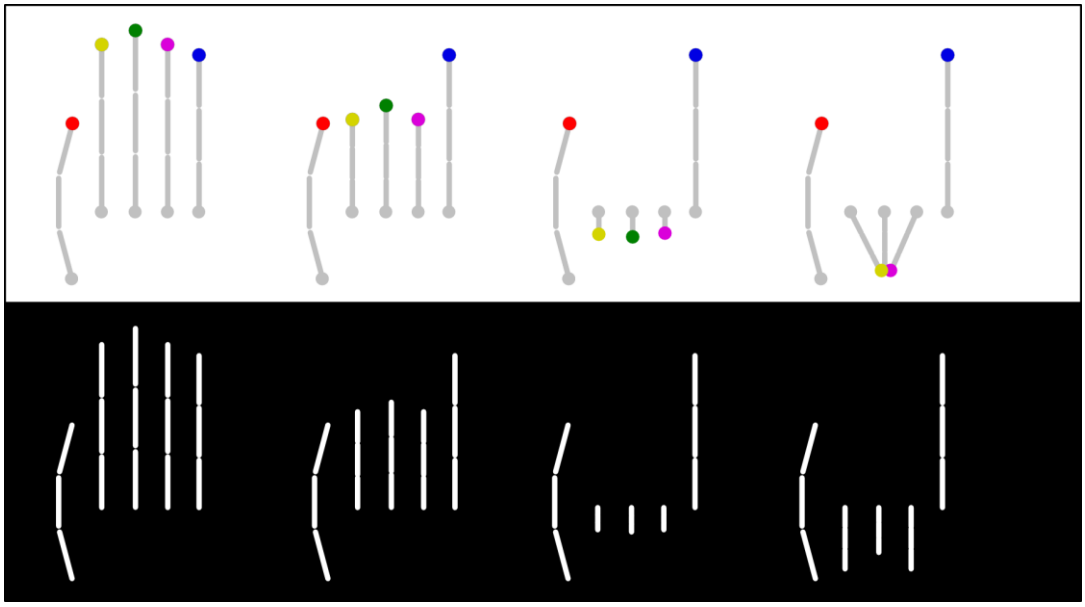

Figure 4.17: At the Top Four Real Articulations of the Fingers Index, Middle and Ring, at the Bottom Their Corresponding Articulation Models

According to the method proposed in this thesis, if one color marker could not be captured in a frame, the finger that the color marker attached is modeled according to the last known coordinates. In the Figure 4.17 it is can be seen clearly that the middle finger is obstructed by the index and ring fingers, so it is

modeled according to its last known coordinates, which does not represent the real articulation.

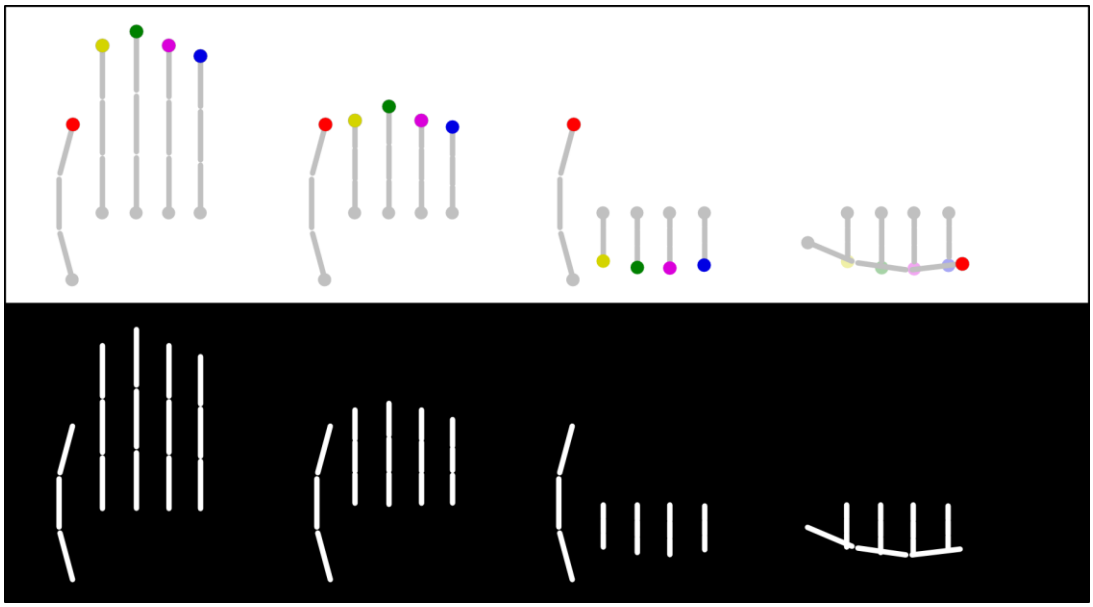

Figure 4.18: At the Top, the Real Articulation of the Fingers, at the Bottom Their Corresponding Model

In the Figure 4.18 four real articulations and their corresponding models are show. Although the thumb obstructs the other four fingers, the articulation shows that the occlusion not necessarily hinders the model's accuracy.

4.2 Parts to be Enhanced

There may be improvements like to integrate the tools which are developed for this thesis into one application/software or handling the starting position and so on.

4.2.1 Hue saturation values capturing tool

The hue and saturation values of the color markers are acquired by a tool. This small software may be removed and the values of hue and saturation can be acquired on the fly. This can be done in the first few seconds of the interaction by detecting the background and asking the user to put the hand in front of the cameras, than segmenting out the background leaving the hand with the color markers attached to the finger tips. Finding the finger tips and calculating the

color markers' hue and saturation values can be done by this or any other approach.

4.2.2 Predefined starting position of hand

The hands starting position is predefined. But in real life application this may be unobtainable or undesirable, therefore the implementation should adapt to other positions of the hand rather than a predefined one.

4.2.3 Finger's Degree of Freedom

The four fingers from index to little finger are assumed not to be capable of moving in the x-axis, this should be available in a real life application.

4.2.4 Assumptions

There are assumptions of the DOF of the fingers and the thumb's articulations; these may adversely affect the resulting hand model. If possible the assumptions should be removed or should be more limited.

4.2.5 Hand may be moving itself

The hand is assumed to be at a fixed position, which may not be desirable for a real life application, the movements of the hand should be handled.

CHAPTER 5

CONCLUSION

There are lots of research done regarding the human hand articulation and the modeling of the 2D/3D human hand in the Human Computer Interaction field. Each have their superiorities and down sides; they try to reveal an easier, familiar, and readily available, if it is possible cheaper and faster way for interaction between human beings and the computers. In short, these researches and their real life implementations are explorations of a better way, sometimes in software sometimes in hardware. There are problems like CPU and time consumptions or costly and unobtainable hardware. These problems can be addressed with two cheap web cameras, a low cost laptop and less resource-consuming software. With this kind of a setup, a human hand can be modeled in a 3D manner. Although it may not be accurate as a hardware system, the proposed method leads to a faster and cheaper solution. In this thesis, the cameras are calibrated, color markers are captured and tracked, articulations calculated and 3D hand model is created.

5.1 Future Work

The 3D hand articulations should be combined with the whole arm articulations for complex human computer interactions. And only one hand is not sufficient for some tasks that are already available in real life problems, therefore both hands and their arms also should be modeled. For the fast startup time with zero calibration the color markers' color values should be a standard. And of course there should be sample implementations like file managers, desktop environments or games.

REFERENCES

[1] **BOOTH, P.A.** (1989), *An Introduction To Human-Computer Interaction*, Lawrence Erlbaum Associates Ltd., East Sussex.

[2] **HEWETT, T.T.** et. al. (1996), *Association for Computing Machinery*, Inc., New York.

[3] **RASKIN, J.** (1994), Intuitive Equals Familiar, *Communications of the ACM*, 17. Vol. 37:9.

[4] **SLATER, B. R.** et. al. (1983), Industrial Process Control System, United States Patent 4413314.

[5] **EDWARDS, W. K.** et.al. (2010), The Infrastructure Problem in HCI, *CHI '10 Proceedings of the 28th International Conference on Human Factors in Computing Systems*, Atlanta.

[6] **BRADSKI, G., KAEHLER, A.** (2008), Projection and 3D Vision, *Learning OpenCV*, O'Reilly Media, Inc, Sebastopol, 405-458.

[7] **OHTA, Y.** et al. (1980), Color Information for Region Segmentation, *Computer Graphics and Image Processing*, 222–241, Vol. 13:3.

[8] **HILLEBRAND, G.** et. al. (2006), Inverse Kinematic Infrared Optical Finger Tracking, *9th International Conference on Humans and Computers (HC 2006),* Aizu, Japan.

[9] http://www.gnu.org/gnu/linux-and-gnu.html

[10] http://gcc.gnu.org/

[11] **BRADSKI, G., KAEHLER, A.** (2008), Overview, *Learning OpenCV*, O'Reilly Media, Inc, Sebastopol, 1-15.

[12] http://www.vtk.org/

[13] **BRADSKI, G., KAEHLER, A.** (2008), Camera Models and Calibration, *Learning OpenCV*, O'Reilly Media, Inc, Sebastopol, 370-404.

[14] **FISHER, R. B., KONOLIGE, K.** (2008), Range Sensors, *Springer Handbook of Robotics*, eds. Bruno Siciliano, OussamaKhatib, Springer-Verlag, Berlin, 524.

[15] **GONZALES, R.C., WOODS, R.E.** (2008), Filtering in the Frequency Domain, *Digital Image Processing*, Pearson Education, Inc., New Jersey, 290.

[16] **TSANG, P.W.M., TSANG, W.H.** (1996), Edge detection on object color, *IEEE International Conference on Image Processing*, Lausanne, Switzerland, 1049–1052.

[17] **TEPICHIN E.** et. al. (1995), Hue, Brightness, and Saturation Manipulation of Diffractive Colors, *Optical Engineering*, 2886–2890, Vol. 34:10.

[18] **KIM, K.M.** et. al. (1996), Color image quantization using weighted distortion measure of HVS color activity, *IEEE International Conference on Pattern Recognition*, Lausanne, Switzerland, 1035–1039.

[19] **BRADSKI, G.R.** (1998), Computer Vision Face Tracking For Use in a Perceptual User Interface, *Intel Technology Journal*, Q2, Vol. 2:2.

[20] **FOXLIN, E., HARRINGTON, M.** (2000), WearTrack : A Self - Referenced Head and Hand Tracker, for Wearable Computers and PortableVR, *Proceedings of International Symposium on Wearable Computers (ISWC 2000)*, Atlanta.

[21] http://www.vrealities.com/glove.html

[22] **MOHR, D., ZACHMANN, G.** (2009), Continuous Edge Gradient-Based Template Matching For Articulated Objects, IfI Technical Report Series, IfI-09-01, Institut für Informatik, Technische Universitat Clausthal, Germany.

[23] **ROSALES, R.** et. al. (2001), 3D Hand Pose Reconstruction Using Specialized Mappings, *Proceedings IEEE International Conference on Computer Vision* (ICCV), Canada.

[24] **BUSS, S. R.,** (2009), Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods, Department of Mathematics University of California, San Diego.

[25] **LANDER, J.** (1998), Making kine more flexible, *Game Developer Magazine*, 15–22,Vol. 11.

[26] **GARG, P.** et al. (2009), Vision Based Hand Gesture Recognition, *World Academy of Science, Engineering and Technology*, 972–977, Vol. 49

[27] **YEH, C.** et. al. (2010), Vision-Based Virtual Control Mechanism via Hand Gesture Recognition, *Journal of Computers*, 1-56, Vol. 21:2

[28] **MITRA, S., ACHARYA, T.** (2007), Gesture Recognition: A Survey, *IEEE Transactions on Systems, Man, and Cybernetics – Part C*, 311-324, Vol. 37:3.

APPENDIX A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Mogol, Ali Can
Nationality: Turkish (TC)
Date and Place of Birth: 4 May 1980, Balıkesir
Marital Status: Married
Phone: +90 312 326 45 23 / +90 535 283 46 74
Email: alicanmogol@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
BS	Hacettepe Univ. Food	2007
High School	BTOL, Balıkesir	1998

WORK EXPERIENCE

Year	Place	Enrollment
2009-Present	Innova A.Ş.	Application Development Consultant
2007-2009	Positive Ltd.	Application Development Specialist

FOREIGN LANGUAGES

Advanced English

HOBBIES

AI/Robotics, History, Movies, Sports