HAND GESTURE RECOGNITION SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRAH GİNGİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2010

Approval of the thesis:

HAND GESTURE RECOGNITION SYSTEM

submitted by **EMRAH GİNGİR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Gözde Bozdağı Akar _____
Supervisor, **Electrical and Electronics Department, METU**

Assoc. Prof. Dr. Mehmet Mete Bulut _____
Co-supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Department, METU _____

Assoc. Prof. Dr. Mehmet Mete Bulut
Electrical and Electronics Department, METU _____

Assoc. Prof. Dr. Aydın Alatan
Electrical and Electronics Department, METU _____

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics Department, METU _____

MSc. Burcu Kepenekci
Paranavision _____

Date: _____

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: EMRAH GİNGİR

Signature :

ABSTRACT

HAND GESTURE RECOGNITION SYSTEM

Gingir, Emrah
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Gözde Bozdağı Akar
Co-Supervisor : Assoc. Prof. Dr. Mehmet Mete Bulut

September 2010, 78 pages

This thesis study presents a hand gesture recognition system, which replaces input devices like keyboard and mouse with static and dynamic hand gestures, for interactive computer applications. Despite the increase in the attention of such systems there are still certain limitations in literature. Most applications require different constraints like having distinct lightning conditions, usage of a specific camera, making the user wear a multi-colored glove or need lots of training data. The system mentioned in this study disables all these restrictions and provides an adaptive, effort free environment to the user. Study starts with an analysis of different color space performances over skin color extraction. This analysis is independent of the working system and just performed to attain valuable information about the color spaces. Working system is based on two steps, namely hand detection and hand gesture recognition. In the hand detection process, normalized RGB color space skin locus is used to threshold the coarse skin pixels in the image. Then an adaptive skin locus, whose varying boundaries are estimated from coarse skin region pixels, segments the distinct skin color in the image for the current conditions. Since face has a distinct shape, face is detected among the connected group of skin pixels by using the shape analysis. Non-face connected group of skin pixels are determined as hands. Gesture of the hand is recognized by improved centroidal profile method, which is

applied around the detected hand. A 3D flight war game, a boxing game and a media player, which are controlled remotely by just using static and dynamic hand gestures, were developed as human machine interface applications by using the theoretical background of this study. In the experiments, recorded videos were used to measure the performance of the system and a correct recognition rate of ~90% was acquired with nearly real time computation.

Keywords: Human Machine Interaction, Hand Gesture Recognition, Face Detection, Skin Color Modeling , Machine Vision

ÖZ

EL İŞARETİ TANIMA SİSTEMİ

Gingir, Emrah
Yüksek Lisans, Elektrik ve Elektronik Mühendisliğ Bölümü
Tez Yöneticisi : Prof. Dr. Gözde Bozdağı Akar
Ortak Tez Yöneticisi : Doç. Dr. Mehmet Mete Bulut

Eylül 2010, 78 sayfa

Bu tez çalışması etkileşimli bilgisayar uygulamalarındaki klavye ve fare gibi girdi çevre birimleri yerine statik ve dinamik el işaretlerini kullanan bir el işareti tanıma sistemini sunmaktadır. Bu tip sistemlere ilgi artmış olmasına rağmen günümüzdeki çalışmalarda hala kısıtlayıcı bir takım kıstaslar vardır. Çoğu uygulama, sadece belirli ışık koşullarında çalışabilme, belirli bir kamera tipi ile çalışabilme, kullanıcının renkli bir eldiven giymesi ya da çok fazla eğitici veriye ihtiyaç duyma gibi çeşitli kıstaslara sahiptir. Bu çalışmada anlatılan sistem, tüm bu kısıtlamaları ortadan kaldıran ve kullanıcıya kendi kendine uyarlanabilen, zahmetsiz bir uygulama sunmaktadır. Çalışma farklı renk uzaylarının ten rengi çıkarma performanslarını karşılastıran bir analiz ile başlamaktadır. Bu analiz, çalışan sistemden bağımsız bir çalışmadır ve sadece renk uzaylarını daha yakından tanımak adına yapılmıştır. Çalışan sistem el tespit etme ve el işareti tanıma olarak iki kısımdan oluşmaktadır. El tespit etme kısmı, düzgelenmiş RGB renk uzayındaki ten rengi gezingeninin görüntüdeki ten rengi piksellerini kabaca eşiklemesiyle başlar. Ardından sınırları, eşiklenen bu ten rengi piksellerinden kestirimlenen, uyarlanmış bir ten rengi gezingeni mevcut koşulların ten rengini çıkarır. Yüzün sabit bir şekli olduğu için, ten rengi piksellerinin arasından biçim analizi ile yüz tespit edilir. Geri kalan bağlantılı ten pikselleri ise el olarak tespit edilir. El işareti ise iyileştirilmiş merkezi

kesit çıkarma yönteminin tespit edilen elin etrafına uygulanması ile tanınır. Tanınan el işareti insan-bilgisayar etkileşimli uygulamalarda klavye ve fare yerine kullanılır. 3 boyutlu uçak savaş oyunu, boks oyunu ve video oynatıcısı uygulamaları bu çalışmanın teorik altyapısını kullanan örnek insan-bilgisayar etkileşimli uygulamalar olarak geliştirilmiştir. Deneylerde, kaydedilmiş video görüntüleri sistemin performansını ölçmek amacıyla kullanılmış ve yaklaşık %90'lık bir doğru tanıma başarısı gerçek zamanlıya yakın bir hesaplama ile elde edilmiştir.

Anahtar Kelimeler: İnsan Makine Etkileşimi, El İşareti Tanıma, Yüz Sezimi, Ten Rengi Modelleme, Bilgisayarla Görme

ACKNOWLEDGMENTS

I express my sincere appreciation to my thesis supervisor Prof. Dr. Gözde Bozdağı Akar and co-supervisor Assoc. Prof. Dr. Mehmet Mete Bulut for their guidance, insight and elegant attitude throughout the research.

I also thank Assoc. Prof. Dr. Aydın Alatan, Assoc. Prof. Dr. Çağatay Candan and MSc. Burcu Kepenekci who kindly agreed to serve in my thesis examining committee.

I wish to thank my parents Hamiyet and Ertaş Gingir and my brother Veli Gingir for their support, encouragement and confidence throughout the years of my education.

I also thank to my friends Gizem Coşkun, Ferhat Can Gözcü, Aydın Güney, Halil Tongül, Nisa Türel, Ramazan Çetin, Eren Alp Çelik, Ahmet Zor and Olcay Demirörs who have great contribution with their supportive dialogues to finish this study.

I would like to thank to my company ASELSAN and my colleagues for their understanding and I also thank to TÜBİTAK for their financial support during my graduate study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Communication in daily life is performed via the help of vocal sounds and body language. However vocal sounds are the main tool for interaction, body language and facial expressions have a serious support in the meanwhile. Even in some cases, interacting with the physical world by using those expressive movements instead of speaking is much easier. Body language has wide range of activities namely eye expressions, slight change in skin color, variation of the vibrations in vocal sounds etc. But the most important body language expressions are performed using hands. Hand gestures would be ideal for exchanging information in recent cases such as pointing out an object, representing a number, expressing a feeling etc and also hand gestures are the primary interaction tools for sign language and gesture based computer control.

With the help of serious improvements in the image acquisition and processing technology, hand gestures become a significant and popular tool in human machine interaction (HCI) systems. Recently, human machine interfaces are based on and limited to use keyboards and mice with some additional tools such as special pens and touch screens. Although those electro-mechanical devices are well designed for interacting with machines and very ordinary in daily life, they are not perfect for natural quality of human communication. Hand gestures and other body language expressions are thought to replace keyboard and mouse for HCI systems in the near future. Many of the significant information technologies companies have been working on such systems. Main application areas of hand gesture recognition in human machine interface systems are keyboard-mouse simulations, special game play without joysticks, sign language recognition, 3D animation, motion and performance capture systems, special HCI for disabled users etc. Especially special game play and motion and performance capture systems based on hand gesture recognition are being designed and used in industry

today. Also in daily life people usually do not want to touch buttons or touch screens in public areas like screens in planes or buttons in automatic teller machines (ATM) because of hygienic considerations. Hand gestures would be an ideal replacement in that manner.

In this study, a hand gesture recognition system was developed to capture the hand gesture being performed by the user and to control a computer system by that incoming information. Many of such systems in literature have strict constraints like wearing special gloves, having uniform background, long-sleeved user arm, being in certain lightning conditions, using specified camera parameters etc. Such limitations ruin the naturalness of a hand gesture recognition system and also correct detection rates and the performances of those systems are not well enough to work on a real time HCI system. This study aims to design a vision based hand gesture recognition system with a high correct detection rate along with a high performance criteria, which can work in a real time HCI system without having any of the mentioned strict limitations (gloves, uniform background etc)on the user environment. Both academic and commercial world lack such an assertive system and this study intends to fill this gap.

This study is composed of a human computer interaction system which uses hand gestures as input for communication. System is initiated with acquiring an image from a web-cam or a pre-recorded video sequence. Skin color is determined by an adaptive algorithm in the first few frames. Once the skin color is fixed for the current user, lightning and camera parameter conditions, hand is localized with a histogram clustering method. Then a hand gesture recognition algorithm namely centroidal profile extraction method is applied in consecutive frames to distinguish the current gesture. Finally, the gesture is used as an input for a computer application. In brief, the scope of the study is divided mainly in 4 parts.

- Skin Color Modeling
- Hand Segmentation
- Hand Gesture Recognition
- Applications

In general, such interaction systems have two challenges: Hand Detection and Hand Gesture Recognition. One needs to find the hand region first prior extracting gesture information. For

this purpose, skin color is segmented in the current image as first step. Choosing the right color space for skin color segmentation is a crucial point which impacts the performance of the following steps in the algorithm significantly. There are 3 useful color spaces for skin color extraction in literature, namely *normalized-RGB*, HSV and $YCbCr$. These 3 color spaces are similar to HVS (Human Vision System) and luminance feature of the current image can be easily eliminated by using each of them [8]. A test bed was constructed to compare the skin color extraction performances of *normalized-RGB* and $YCbCr$ color spaces. For hand segmentation, *normalized-RGB* is used as the color space in this study. A general skin locus, which extracts all kinds of skin colors under all lightning conditions is used as coarse skin color thresholds and this gives a quick but rough elimination of the non-skin pixels. Remaining pixels are called skin candidate pixels. It is likely to have a high false positive detection rate among skin candidate pixels because coarse skin segmentation thresholds are designed so that all type of skin colors (Asian, European, African etc.) are extracted under all lightning conditions by all camera parameters except extreme cases. At this point, it is needed to eliminate false positives and decide the narrowed skin locus for the current lightning and camera conditions. For this reason, an effective hand segmentation process which is based on a technique used for face detection in a former study [2] is applied on the skin candidate pixels. This process is initiated with a fine skin color segmentation which follows the coarse skin segmentation by extracting $(g - r)$ and $(g + r)$ histograms of skin candidate pixels. A typical PC user is expected to sit in front of the monitor and staring on it. Design of the system is based on this fact so the image acquisition device is attached to the monitor and a clear frontal face image is expected to be in the acquired image. As a consequence, frontal face would yield peaks in $(g - r)$ and $(g + r)$ histograms. Since skin candidate pixels have just skin pixels and skin like pixels, frontal face and hand(s) would correspond to first or second biggest local peaks in the histograms. By considering those 4 local peaks (2 for $(g-r)$ and 2 for $(g+r)$), new narrowed skin color borders are generated. By cross matching of these 4 locals peaks, 4 new narrowed skin loci is extracted and these loci are subjected to the skin candidate pixels and new images are held by using new loci. One of these loci would correspond to true skin locus for the current conditions. To decide the true skin locus, neighboring pixels of each image are grouped to regions. A clear frontal face image is searched in each of the resultant bitwise image by considering height to width ratio, ellipse fitting and facial components properties of each region in an adaptive manner. Once a region is pointed out as the frontal face, skin color thresholds used to construct that face are finalized as new narrowed skin color boundaries.

If the algorithm gives no valuable information to locate face in the image, the current frame will be dropped and this mislead information will be used for the estimation of fine skin locus in the next frame. By this feedback mechanism, frontal face will be located in a few frames and fine skin locus boundaries will be fixed. Frontal face corresponds to head and the biggest remaining connected group(s) of pixels correspond(s) to hand(s) in the image. Once the hand is segmented it is checked that if the user is wearing a short sleeved cloth or not. If a short sleeved cloth is in question, then the arm would be visible to the camera and wrist detection procedure is applied on the segmented region. Details of the hand segmentation is in chapter 4.

Once the hand is segmented clearly in the current image, gesture recognition process is started around the segmented hand. Many techniques are searched in literature for gesture recognition and a vision based rotation invariant method was chosen for this purpose. Other methods are discussed in chapter 2. The proposed method for gesture recognition in this study is called centroidal profile extraction and adding important modifications to the mentioned methods in literature [5], [7]. According to the centroidal profile extraction method, growing circles are drawn around the mid point of the hand-wrist intersection line. Each circle is considered as a contour to move on and a polar transformation is used to count the number of fingers being shown. If a point on a circle is a skin pixel then the corresponding angle on the *Number of Skin Pixels vs. Angle* graphic will be increased by one. Skin pixels vs Angle histogram of this growing circles are extracted and peaks on that histogram are counted. Number of peaks in this histogram will give the number of fingers being shown to the camera. In the scope of this thesis, hands are assumed to be upwards (as a typical PC user would do). So just the $0-180^\circ$ or $180-360^\circ$ intervals are considered according to the starting point and direction of the circle contour. Since the number of fingers being shown is assigned as the input for the HCI system, system becomes rotation invariant for the given rotation angle interval. Also since the hand is localized by the hand segmentation in the previous step the system is also translation invariant. Details of the hand gesture recognition procedure is in chapter 5.

Entire system was tested in a test-bed to take out statistical results to measure the success of the system. Pre-recorded videos were analyzed frame by frame if the input gesture can be recognized correctly or not. The results were compared to the results of previous studies. Detailed results and comments about the entire study are in chapter 7.

Finally, all the mentioned procedures were implemented by MATLAB in a two cores computer. The system is aimed to work in real time. In the calibration process of skin color, the system works 2-3 frames per second but once the skin calibration is settled gesture recognition procedure works nearly 10 frames per second. Also, some applications using the mentioned gesture recognition algorithm were implemented. A boxing and a 3D flight war games were designed which are entirely played by hand gestures. Also a movie player and a simulation of an ATM machine which are controlled just by hand gestures are functioning. All these systems were constructed in Visual Studio 2005 using C# programming language and the system working in the background to recognize the hand gestures is working on MATLAB. Details about the applications based on the discussions of the thesis are in 6.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Many researchers have proposed numerous methods for hand gesture recognition systems. Generally, such systems are divided into two basis approaches namely $glove - based$ and $vision - based$ approaches. In glove based analysis, detection of the hand is eliminated by the sensors on the hand and 3D model of the hand is easily subjected to the virtual world and analysis comes next. Such systems are optimal for body motion capture purposes and widely used in industry. On the other hand vision-based analysis are more natural and useful for real time applications. A healthy human can easily identify a hand gesture, however for a computer to recognize hand gesture first the hand should be detected in the acquired image and recognition of that hand should be done in a similar way as humans do. Yet this is a more challenging approach to implement because of the limitations of such a natural system. The vision-based approaches are carried out by using one or more cameras to capture and analyze 2D or 3D shapes of hands.

Detection and gesture analysis of the hands is a growing literature topic and has many user environment limitations for most of the studies. Segmentation of the hand is the first step of such systems. Exceptionally, such systems in [10], [12] and [13] gloves or finger marks have been used to extract the hand posture information in the frame and ease the hand segmentation process by eliminating the varying skin color issue of the problem. This technique allows the system to detect hands in a straightforward manner and it is more robust to change in lightning conditions and it is also independent of the user's skin color. Another easiness is carried out running the hand gesture recognition system in front of a simple and uniform background like

a black curtain [11]. Such systems need to distinguish skin color but since background could be estimated easily, it would be very easy to segment the hand region from the background. On the other hand, such systems involving gloves or uniform background ruin the natural behavior of gesture applications by limiting the user environment. Since the scope of the thesis is detecting hands in a complex background and since the hand has no strict shape that can be easily identified in an image, system is calibrated by using face color information. Face has a strict shape and detection of face is much more straightforward. Consequently, one can say that the hand detection is based on a face detection algorithm.

Face detection methods can be categorized into 3 main sections: *Feature Invariant Methods*, *Template Matching Methods* and *Appearance − Based Methods* [14]. Feature invariant methods are capable of localizing faces in an image even if pose, viewpoint or lightning conditions vary. Template matching methods are based on storing several standard patterns of the face and correlating them with the current input image to detect faces. Lastly, appearance-based methods (in contrast to template matching) face patterns are learned from a set of training images which include representative variability of facial appearance. Categorization of face detection methods and instance studies of each method are summarized in table 2.1.

In recent years, with the introduction of a new approach [33], which has a high detection rate, new studies are mostly concentrated on Boosting and HMM. The most tempting side of using those methods is that they usually work with grayscale images instead of colored images and thus it eliminates the drawbacks of such color based noise issues. This innovative approach is using a well-known technique namely Adaboost classifier which was mentioned in [32]. Adaboost classifier is an effective tool to select appropriate features for face detection. This feature extraction technique does not need skin color information and have less computation time with the use of integral image concept. But the drawback of this method is that it requires a training process. This process often needs huge-sized sample images to have a high detection rate. Sample images would have thousands of positive images (include face) and thousands of negative sample images (don't include face). Also training process would have a high computation time and it might be several days to complete training.

By considering the drawbacks of training based methods and since there is a huge amount of studies based on training in literature, a feature invariant method was chosen to locate faces in this study. Also the main of this thesis study is recognizing hands and detection of face is just

Table 2.1: Detection Methods

Approach	**Instance Study**
Feature Invariant Methods	
• Facial Features	Facial Components Analysis [21]
• Texture	Gray Scale Texture Classification [23]
• Skin Color	Adaptive Gaussian Mixture Model [16]
• Multiple Features	Skin color, shape analysis and facial components. [2]
Template Matching Methods	
• Predefined Templates	Shape Template Matching [35]
• Deformable Templates	Skin Active Shape Model for Face Alignment. [20]
Appearance-Based Methods	
• Principal Components Analysis	Event Detection by Eigenvector Decomposition [17]
	Eigenface Decomposition for Face Recognition[34]
• Neural Network	Motion Pattern Classification by Neural Networks [15]
• Support Vector Machine (SVM)	SVM with Fisher Kernels [12]
• Bayes Classifier	Dynamic BN for Gesture Recognition [18]
• Hidden Markov Model (HMM)	Input-Output HMM for Gesture Recognition [19]
• Boosting and Ensemble	Detection using Boosted Features.[22]

an intermediate tool. Investigation of a new technique for face detection would be tempting and Multiple Features in table 2.1 was a good choice for detection of face. Skin color is the central invariant feature of this study. It is indicated that human skin color is independent of human race and the wavelength of the exposed light [36]. This fact is also valid for the transformed color spaces of common video formats and thus skin color can be defined as a *globalskincolorcloud* in the color space and this cloud is called skin locus of that color space [3]. The thresholds of the skin locus is too large to extract current skin color in an input image correctly. Since shadows, illumination and pigmentation of human skin conditions would vary in a wide range, it is reasonable to adapt this general thresholds and narrow the skin locus for the current conditions. Skin locus is supposed to include all the skin pixels in an image with some other skin like pixels. Those false positive pixels should be eliminated by a fine skin segmentation.

According to face detection method introduced in [2], colored images are investigated in 2 steps namely coarse skin color segmentation and fine skin color segmentation. For coarse skin color segmentation fixed skin color thresholds in *nRGB* color space are used (skin locus).

Many studies in literature use different skin locus thresholds for different color spaces to locate faces or hands in images. [25] starts with an RGB image and by apply a dimension reduction algorithm to propose its own skin locus in two dimension to compare its performance with HSV skin locus. [24] and [29] compare HSV/HSI, RGB, TSL and $YCbCr$ color space skin locus performances. [26] starts the segmentation of skin color in YUV color space to have a a quick result and then tune the current skin color with a quasi-automatic method which needs some user input. In [27], chrominance along with luminance information in $YCbCr$ color space was used to segment the skin color for current conditions and histogram clustering is used for fine skin segmentation.

In feature invariant face detection methods skin segmentation is followed by face verification. Once the skin color is segmented in the obtained image, skin pixels are grouped by a region growing algorithm to extract blobs. Those blobs are investigated if they are face or not. Many methods have been proposed for this purpose in literature. The most simple one is that calculating height to width ratios of the segmented blobs [31]. Typically a frontal face's height to width ratio would be in a certain interval and this information would yield elimination of some blobs which are not definitely corresponding to real face in the image. Height to width ratio alone is not sufficient to detect the face blob because skin like pixels might still construct similar height to width ratio blobs. in [28] symmetry analysis and facial components analysis are introduced to detect faces. Detecting symmetric eyes, lips and nose would be the best method to recognize detect a face however it contributes an important computation time to the algorithm. [2] propose a method between the simplicity and the computational complexity of these two methods and presents *blob's mismatch area* method. According to this method, an ellipse is fitted to each face candidate blob and best fitted ellipse is pointed out as the actual face.

Finding out hands directly would not be an effective way since hands do not have a strict shape. Once the face is detected, the other skin pixel blobs can be supposed as hands. Hand gesture process is started at this point. For hand gesture recognition part of this study, HMM or Adaboost classifier type training based methods have very limited usage because of the non-strict structure of the hand. To recognize a gesture once need to train positive images (include the defined gesture) and negative images (do not include the defined gesture). But negative images have a serious role at this point. Since many hand poses might yield similar training data, reliance to the training data would be limited. So an adaptation of a well-known

hand gesture recognition method was implemented in this study. According to the proposed methods in prior studies [7] and [5] centroidal profile extraction of the hand is extracted around the center of the palm and histogram clustering is applied to the resultant data to recognize the gesture. Such an algorithm would typically count the number of fingers being shown to the camera. this can capture 6 gestures for each hand namely 1, 2, 3, 4 or 5 fingers or no fingers(punch) conditions. If the algorithm is used for two hand gesture recognition $6x6 = 36$ gestures can be recognized by using the mentioned method. In this thesis, an adaptation of that method is proposed and a higher correct detection rate is provided.

There is a limited amount of studies in literature for the hand gesture recognition. Recognition methods, like in the detection procedure, are mainly rely on algorithms which need training or different environmental constraints. A clear summary of such algorithms are shown in table 2.2.

Table 2.2: Gesture Recognition Methods

Reference	Primary Method of Recognition	Number of Gestures Recognized	Background to Gesture Images	Additional Markers Required	Number of Training Images
39	Hidden Markov Models	97	General	Multi-colored gloves	400
5	Entropy Analysis	6	No	No	400
41	Linear approximation to non-linear point distribution models	26	Blue Screen	No	7441
42	Finite State machine modeling	7	Static	Markers on glove	10 sequences of 200 frames each
43	Fast Template Matching	46	Static	Wrist band	100 examples per gesture

In literature, if a gesture recognition system is constructed on a training based methods, then the number of gestures that will be recognized could be increased. However if it is an invariant

method or state based method the number of gestures could not be increased easily. However, the system would be faster and independent of the training process.

CHAPTER 3

SKIN COLOR MODEL

3.1 Introduction

Previous studies clearly indicates that people width different skin colors can be modeled by a skin locus in different color spaces. The modeling is done by collecting different skin color values, from different users under different lightning conditions and a meaningful distribution is tried to be attained among those values. There are studies in literature to extract significant skin color boundaries to segment skin pixels in a given image effectively. These boundaries were designed to include all possible skin color values and named as skin locus. For HSV color space, a pixel is classified as skin if the following conditions are satisfied [9].

$$0 < H < 50 \tag{3.1}$$

$$0.23 < S < 0.68 \tag{3.2}$$

Also the $YCbCr$ skin color thresholds were as the following [9].

$$80 < Y \tag{3.3}$$

$$85 < Cb < 135 \tag{3.4}$$

$$135 < Cr < 180 \tag{3.5}$$

And the last skin color thresholds were for the *normalized-RGB* color space skin locus [2].

$$g \leq r \tag{3.6}$$

$$g \geq r - 0.4 \tag{3.7}$$

$$g \geq -r + 0.6 \tag{3.8}$$

$$g \leq 0.4 \tag{3.9}$$

As clearly seen in the above equations, the thresholds for the experiments are fixed. Each equation represents a border line for the skin locus in the corresponding color space. But which one segments skin color more effectively is a significant question. Most of the studies in literature just give the skin locus thresholds, however the distribution of the skin pixels' color information would be distinctive to choose the right color space for the current application. For instance, *n-RGB* color space skin locus may cover a bigger area than *YCbCr* skin locus but for some current conditions the distribution of *n-RGB* color space skin locus may be concentrated in a narrow region. In that case, for certain conditions one can interpret which color space to use if the skin color modeling of each color space is known. This chapter models the skin locus of *n-RGB* and *YCbCr* color spaces and compares their features to give an answer to mentioned questions.

3.2 Properties of Skin Color Models for Effective Skin Detection

Skin color segmentation is of utmost importance in a real time hand gesture system. Once the skin color for the current conditions is extracted effectively, working on the segmented regions would be easier and faster and accordingly system would have higher correct detection rate. Skin color model should be robust against environmental changes such as changing in the lightning conditions or changes in camera parameters and it should also work with users with different skin colors. Prior studies have shown that human skin color is independent of human race and the wavelength of the exposed light [36]. This observation introduces a requirement that the color space should be able to remove the luminance feature in an effective way. Pure color information should be obtained and it must be independent of the brightness of the scene. Intensity changes might occur due to changes in light source quality or changes in geometrical issues, e.g., distance from the light source. On the other hand, chrominance changes are usually due to the different spectral composition of light sources, e.g., daylight, fluorescent light or tungsten light [7].

A frequently used method to have a robust system against changes in intensity is to transform *RGB* color space into a color space where chrominance and luminance components are orthogonal. But the pure color information namely chrominance values of skin color are different in different color spaces and skin locus performances of each space might vary. As mentioned in chapter 2, there is a skin locus for each color model. The skin locus should

cover all kinds of skin colors (European , African, Asian etc.). On the other hand it should cover a small region in the color model map to yield smaller amount of false positive results. One must consider all these facts to choose the right skin color model for detection and recognition of hand gestures. There are several color spaces some of which are named as; *RGB*, HSV, *Normalized-RGB*, YUV and $YCbCr$. Most of the studies in image processing area have been assimilated in the YUV or *RGB* color spaces because TV, webcam and pre-recorded video data are usually available in these color spaces. Although other color spaces aim to percept the colors in a more uniform and accurate way as the human perceptual system, transformation of video signals to such color spaces is a very time consuming method. It is a trade of between computation time and correct detection rate for the system in this study and the performances of these color spaces need to be compared.

Robustness is achieved if a color space perfectly separating the chrominance from luminance component. HSV, $YCbCr$ and *Normalized-RGB* color spaces apart the chrominance and luminance values and make them to be worked on independently. However this separation leads to a success in skin color segmentation just up to a point. Some valuable skin color information would be lost by the transformation of color space and by choosing just chrominance component to work on. It is exploited that, intensity component provides substantial information on the segmentation of skin and non-skin pixels in an image and thus absence of illumination does not help boost performance [37]. There are studies in literature considering this fact and developing such systems using both chrominance and luminance information [30]. On the other side, it is just another issue to trade off between losing some pixel information and having a definable skin locus to easily work on and having low computational complexity in the algorithm. The scope of this thesis study involves just chrominance components of orthogonal color spaces and compares their robustness in the manner of CD (correct detection) and CR (correct rejection).

3.3 Modeling the Skin Color in Different Color Spaces

To experiment the effect of changes in spectrum of the light source and to see if ignoring luminance component will work to extract a mathematically definable skin locus, a training test-bed was constructed and *RGB* values of skin pixels in 6 different videos have been recorded. Each video was recorded in a different environment with different lightning condi-

tions and with 2 different cameras. Instance of those videos are shown in figure 3.1.

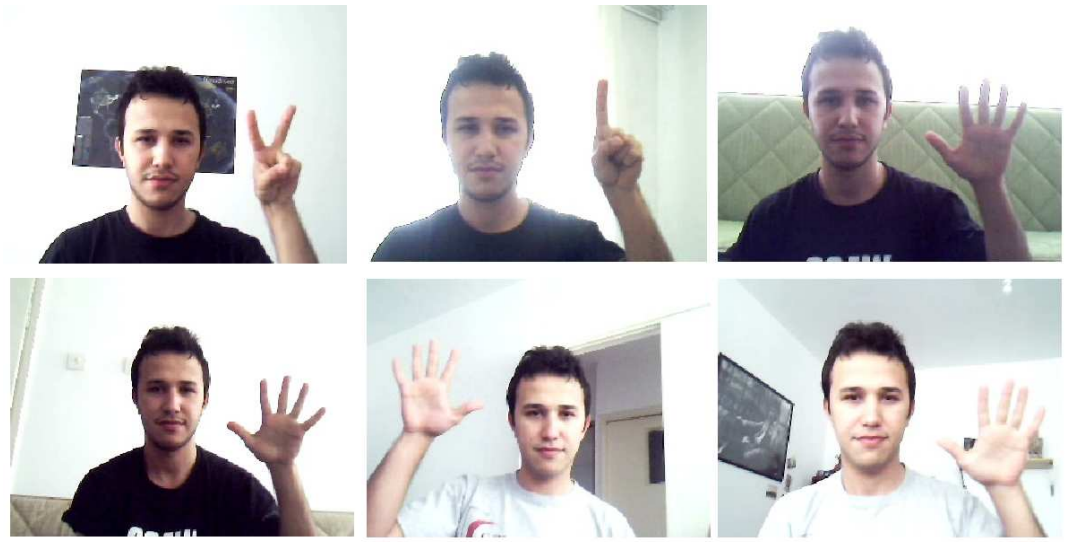


Figure 3.1: Video instances of training videos. Each video has been captured in different lightning conditions. In images from left to right top to bottom: Lightning is subjected from Left, Front and Back, Back, Right, Right and Front, Front.

Normalized-RGB, HSV and $YCbCr$ color spaces were used to design the skin color models and compare the performance of those models. 69823 skin pixels were recorded to analyze and 4504 of them were purely white pixels where $Red = 255$, $Green = 255$, $Blue = 255$ and 9 of them were purely black pixels where $Red = 0$, $Green = 0$, $Blue = 0$. Those pixels were discarded in the design process because of the noise characterizes of the camera. In such analyzes, pixels with minimum intensity $I_{min} \leq Noise\ Free\ Upper\ Level$ should be used because chromaticity calculations are unreliable due to the high level of noise in low *RGB* camera responses. Even for high *RGB* camera responses, the color information is distorted if one or some of the elements of a pixel are under extreme illumination. For this reason it is assumed that each channel should be less than 8 bits and pixels having at least one element is 8 bits (255) are ignored directly. Because of the fact that most of the cameras have a non-linear intensity response, at higher *RGB* outputs such pixels around white can be neglected to analyze. Correspondingly, a total of $69823 - 4504 - 9 = 65310$ skin pixels were carried out in the following analysis.

3.4 General Skin Chrominance Model

As in many natural process, random variations of skin chrominance tends to cluster around a mean in the chrominance space. This is the most commonly observed probability distribution case and is called Gaussian (normal) distribution. *H-S* , *g-r*, and *Cr-Cb* components construct chrominance in each color space and it is expected that each distribution of their skin color values would yield a Gaussian distribution. To extract mathematically useful information, the histograms of the chrominance components should be like in the following form.

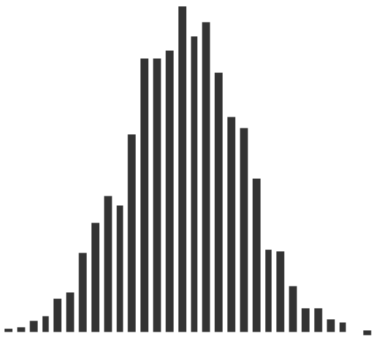


Figure 3.2: Typical Gaussian Distribution

If such a distribution is observed in chrominance data, the mathematical representation would yield the following well known probability density function.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.10}$$

If the two of the chrominance components (e.g. Hue and Saturation) demonstrate such a distribution shown in figure 3.2, skin chrominance distribution of the skin pixels can be modeled by a Gaussian joint probability distribution given by

$$p[\mathbf{x}(i, j/W_s) = (2\pi)^{-1}|\mathbf{C}_s|^{-1/2} \exp\left[-\lambda_s{}^2(i, j)/2\right] \tag{3.11}$$

where the vector $\underline{\mathbf{x}}(i, j) = [\underline{x}(i, j) \;\; \underline{y}(i, j)]^T$ corresponds to values of the chrominance (x, y) (can be thought as H and S or g and r components) of a pixel with coordinates (i, j), W_s is

the distribution representing skin color, C_s is the covariance matrix for skin chrominance, and $\lambda_s(i, j)$ is the Mahalanobis distance from the vector $\underline{\mathbf{x}}(i, j)$ to the mean vector $m_s = [m_{x_s} \; m_{y_s}]^T$ obtained for skin chrominance.

$$[\lambda_s(i, j]^2 = [\; \underline{\mathbf{x}}(i, j) - m_s \;]^T \; \mathbf{C}_s^{-1} \; [\; \underline{\mathbf{x}}(i, j) - m_s \;] \tag{3.12}$$

Equation 3.12 defines elliptical surfaces in chrominance space of $\lambda_s(i, j)$ centered about m_s and where C_s determines the principal axes. Equation 6.1 presents the probability of a pixel with coordinates (i, j) belongs to the class W_s. Correspondingly, if $\lambda_s(i, j)$ (Mahalanobis distance) of a pixel increases, probability of that pixel belonging to class W_s decreases.

By the above theoretical background information, one can model skin locus in a color space by estimating $\mathbf{m}_s$ and $\mathbf{C}_s$. In color space calculations $\mathbf{m}_s$ is constructed by the mean values of the chrominance components (e.g. μ_{Cb} and μ_{Cr}) which were extracted from the recorded skin sample pixels. Then the mean vector of a color space is defined as,

$$m_s = [\mu_x \; \mu_y] \tag{3.13}$$

and mean values of x and y components can be estimated directly by the values of recorded skin sample pixels using the following equations:

$$\mu_x = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{3.14}$$

$$\mu_y = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{3.15}$$

where n is the total number of pixels used in the experiment and x_i and y_i values are the chrominance components' values of the i^{th} pixel respectively in defined color space.

For the estimation of the covariance matrix $\mathbf{C}_s$, let us start with the definition of variance in discrete one dimensional case,

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_x)^2 \tag{3.16}$$

And covariance matrix of the two chrominance components are,

$$C_S = \Sigma[(S - \mu_s)(S - \mu_s)^T] \tag{3.17}$$

where $\underline{\mathbf{S}}$ is the vector composed of the chrominance components,

$$S = \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.18}$$

Finally,

$$C_S = \begin{pmatrix} \sigma_x^2 & C_{xy} \\ C_{yx} & \sigma_y^2 \end{pmatrix} \tag{3.19}$$

Since standard deviations of chrominance components would be real and positive,

$$C_{xy} = C_{yx} \tag{3.20}$$

Finally, cross covariance of the chrominance components and variance of each chrominance component need to be estimated. The estimations in the discrete case are done by the following formulas. And cross covariance of the chrominance components C_{xy} can be estimated by the following formulas.

$$C_{xy} = \sum_{i=1}^{n}[x_i - \frac{1}{n}\sum_{j=1}^{n} x_j][y_i - \frac{1}{n}\sum_{j=1}^{n} y_j] \tag{3.21}$$

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)^2 \tag{3.22}$$

$$\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_y)^2 \tag{3.23}$$

By considering the above equations, one can model distribution of skin color in a defined color space. Here the aim is calculating the *mean* and *variance* of *CrossCovariance* values

of the skin chrominance components and define an ellipse that modeling (circulating) the skin locus. An instance of a distribution with its covariance matrix $\mathbf{C}_S$ and its locus is illustrated in figure 3.3.

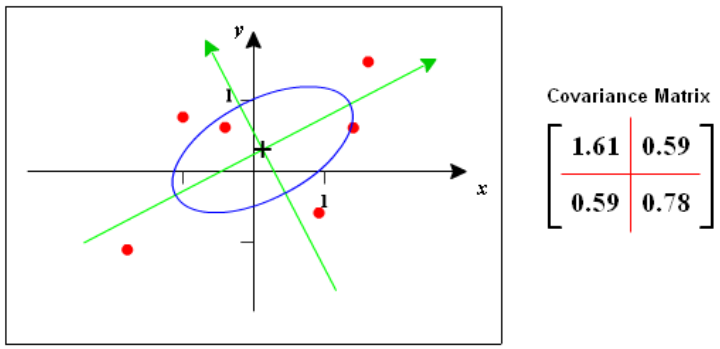


Figure 3.3: An instance of a distribution and visualization of its covariance matrix

3.5 RGB Color Space

The cone-sensors in human visual system are coarsely divided into three regions, namely Red, Green and Blue [1]. So *RGB* color system has been developed which uses these three colors as the main colors and the other colors are described as the combination of these principal colors (Figure 3.4). Since human visual system works in a similar way, early monitors and image acquisition devices were developed using *RGB* color space. In recent years, although other color spaces are well defined to such systems, *RGB* color space is still the most common color space to represent images.

As seen in (Figure 3.4), $[R, G, B] = [0, 0, 0]$ corresponds to black pixels and the $[R, G, B] = [255, 255, 255]$ and other values between these boundaries compose the intermediate colors. The possible number of colors that can be defined by using *RGB* color space is

$$N = (2^{(3)})^{(8)} = 16.777.216 \tag{3.24}$$

which is quite sufficient to display all natural colors that can be identified by human eye.

RGB color space is suitable for hardware design. But it is not linear and luminance feature of the image is completely conveyed. This fact makes RGB color space useless for many

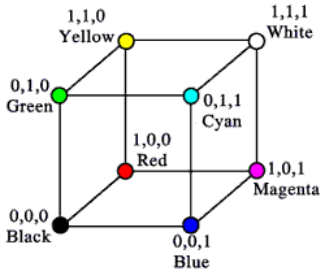


Figure 3.4: RGB Cube

of the image processing applications. To prove this observation, the histogram of the *Red*, *Green* and *Blue* components for 65310 skin pixels are extracted and the results are illustrated in figure 3.5. As clearly seen in each histogram, skin pixels are distributed in the entire color space and they are not concentrated in any region to create a valuable mathematical expression for skin color segmentation.

3.6 Normalized RGB Color Space

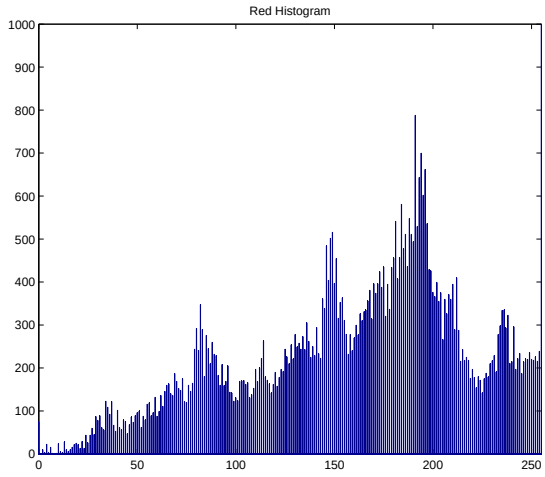
Normalized RGB color space is the color space where *RGB* color information is normalized to overall brightness of the pixel.

$$r = \frac{R}{R+G+B} \tag{3.25}$$

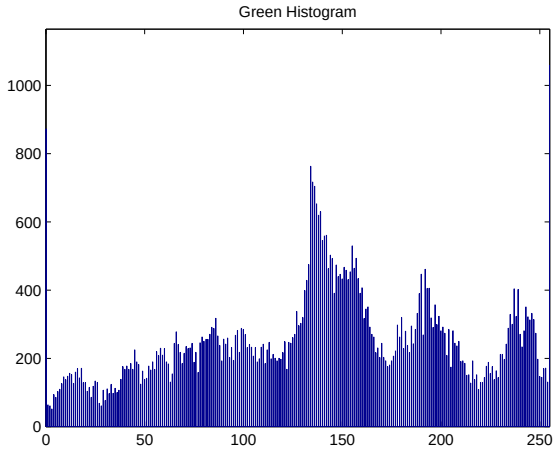
$$g = \frac{G}{R+G+B} \tag{3.26}$$

$$b = \frac{B}{R+G+B} \tag{3.27}$$

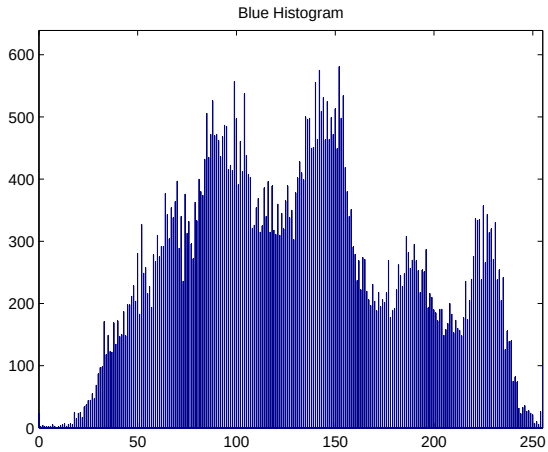
As seen in equations 3.26, 3.27 and 3.27 $(r+g+b) = 1$ which yields just two independent variables. One can infer b as $b = 1-r-g$. This component reduction feature is useful for the upcoming steps of the algorithm in the manner of computation time. This reduction is due to elimination in the luminance feature but it conserves chrominance information. Let us give an example to this feature elimination: $R = 1, G = 1, B = 1$ corresponds to nearly a black pixel in RGB color space and $R = 255, G = 255, B = 255$ corresponds to a white pixel



(a) Red Component



(b) Green Component



(c) Blue Component

Figure 3.5: Histograms of R, G and B components for the skin pixels in training.

accordingly. However they both represents $r = 1/3, g = 1/3, b = 1/3$ in n-RGB color space where r, g and b is the elements of n-RGB color space. At first sight it can be seen confusing that white and near black pixels are leading the same information in n-RGB color space but the chrominance values of those two pixels are exactly the same, which means under certain brightness conditions those two objects can be seen exactly the same to human eye. Since chrominance information is the most valuable color information to identify objects n-RGB color space might be a good choice in hand detection applications.

The training process results of the $n - RGB$ color space are shown in figure 3.6.

Since the histograms of the in figure 3.6 is in the type of Gaussian model illustrated in figure 3.2, the multivariate distribution of this color space can be modeled. To search for the skin locus in this color space first calculate the mean and variance and cross covariance of r and g components in this color space. To calculate the mean values of r and g, estimation shown in equations 3.14 and 3.15 are used and the mean values for these components in training are:

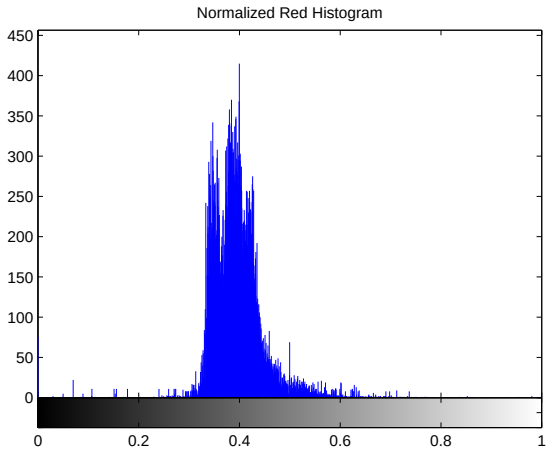
- $\mu_r = 0.3949$
- $\mu_g = 0.3032$

Variance of the components are also calculated by the approach shown in equations 3.22 and 3.23 and the corresponding variance values for those components are:

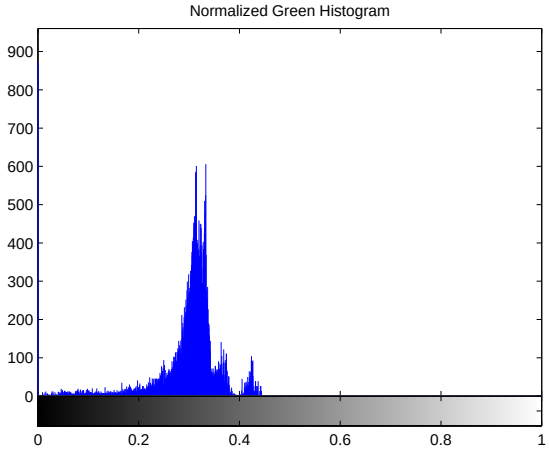
- $\sigma_r^2 = 0.0019$
- $\sigma_g^2 = 0.0032$

Accordingly, bell shaped Gauss curves are fitted to the data of the observations and illustrated in figure 3.7.

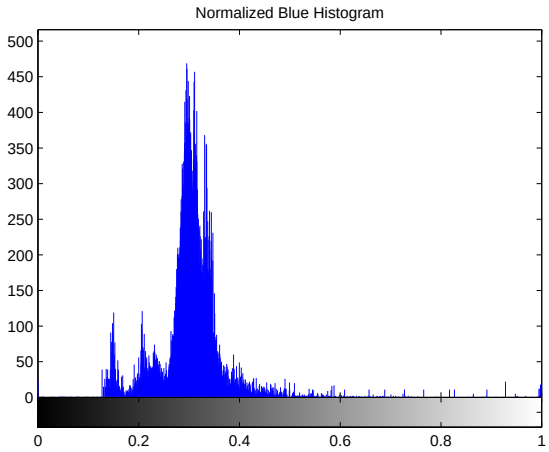
With mean and variance information of each component Gauss distributions of the components are extracted. But with using the covariance information one can interpret the joint pdf distribution of r and g components. To do that one needs to calculate the covariance of those components and by using the information given in section 3.4. Let us first illustrate the distribution of r and g components in $2D$ chrominance space which where held by training



(a) Normalized Red Component

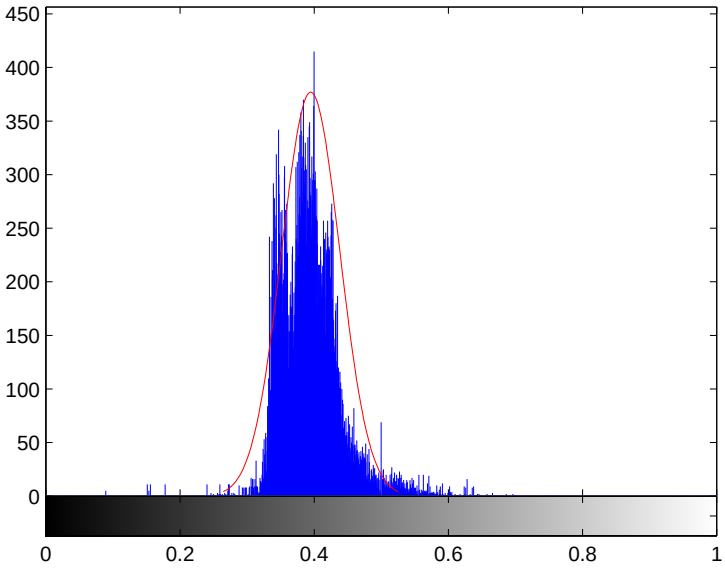


(b) Normalized Green Component

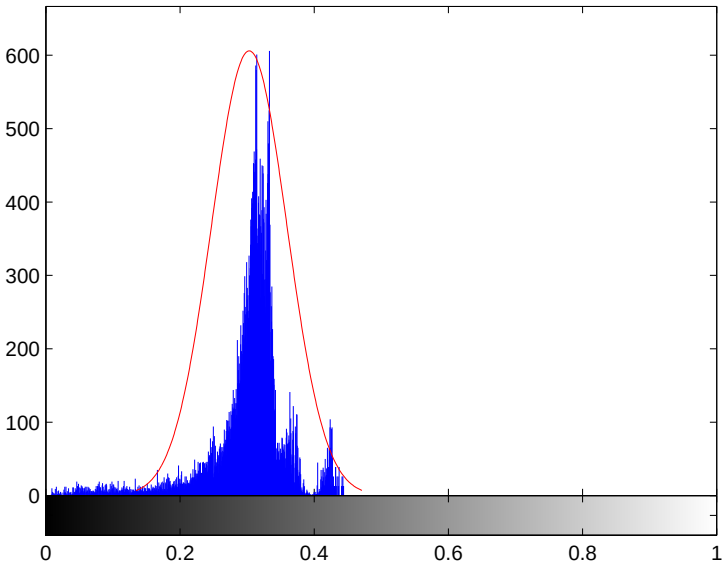


(c) Normalized Blue Component

Figure 3.6: Histograms of r, g and b components for the skin pixels in training.



(a) Gauss distribution of r



(b) Gauss distribution of g

Figure 3.7: Gauss distributions of r and g components

data in figure 3.8. Finally by calculating the covariance between g and r and using the theoretical background mentioned in previous section the elliptical surfaces for the skin locus are extracted and illustrated in figure 3.9.

3.7 YCbCr Color Space

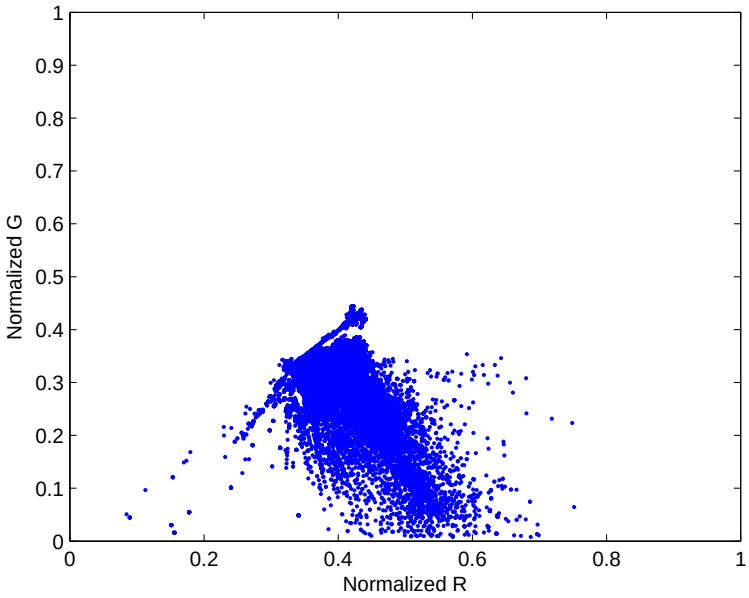
$YCbCr$ color space is another color space where chrominance and luminance values of a given pixel are separated from each other. Y stand for luminance and Cb, Cr values carry chrominance information. This color space is used extensively in digital image world. Especially for image compression applications it is a very useful color space. Transformation from RGB color space to $YCbCr$ is done by the equations 3.29.

$$\begin{aligned} Y &= C1 * R + C2 * G + C3 * B \\ Cb &= (B - Y)/(2 - 2 * C3) \\ Cr &= (R - Y)/(2 - 2 * C1) \end{aligned} \tag{3.28}$$

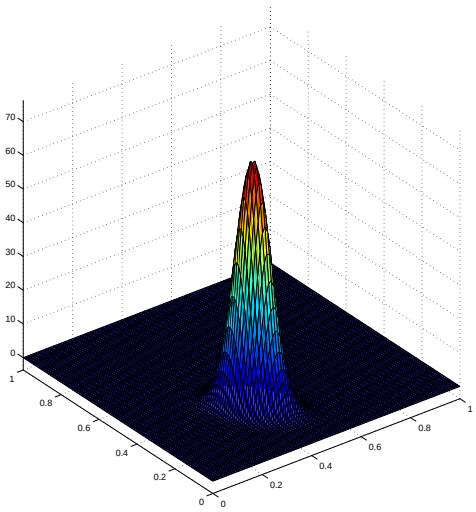
where C1=0.2989, C2=0.5866, C3=0.1145 for standard images and C1=0.2126, C2=0.7152, C3=0.0722 for HD standard images. The transformation from RGB to $YCbCr$ color space is a linear transformation unlike $RGB - HSV$ and $RGB - nRGB$ which means no loss of data occurs in the transformation process.

The visual representations of the $YCbCr$ obtained from the training skin pixel samples are shown in figure 3.10.

By considering the histograms of Cb and Cr components in figure 3.10 one can conclude that those distributions are in the type of Gaussian model illustrated in figure 3.2, and the multivariate distribution of this color space can be modeled. To search for the skin locus in this color space first calculate the mean and variance and cross covariance of Cb and Cr components. To calculate the mean values of Cb and Cr, estimations shown in equations 3.14 and 3.15 are used and the resultant mean values for these components in training are:



(a) Distribution of $(g - r)$ in 2D components



(b) Distribution of $(g - r)$ in 3D components

Figure 3.8: $(g - r)$ distribution of skin pixels in training. Illustrates coarsely skin locus.

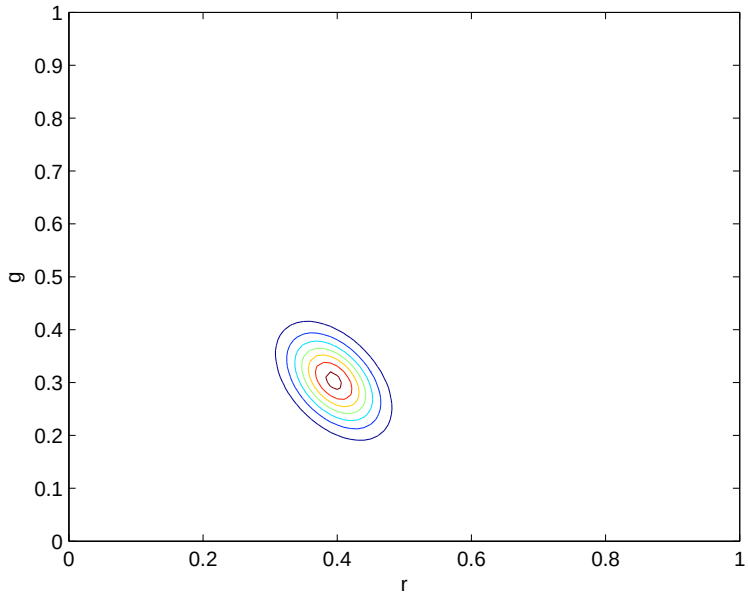


Figure 3.9: Skin Locus in $(g - r)$ chromaticity diagram.

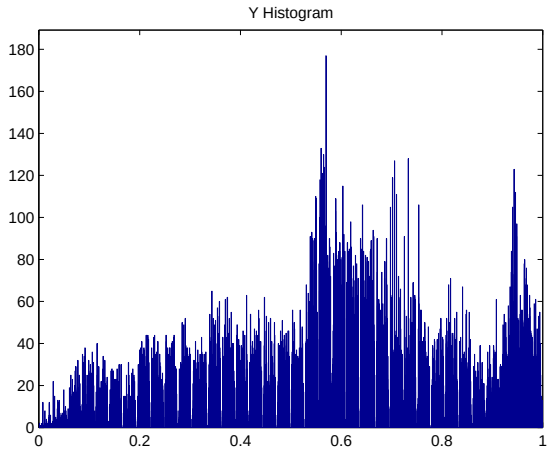
- $\mu_{Cb} = -0.0388$
- $\mu_{Cr} = 0.0638$

Variance of the components are also calculated by the approach shown in equations 3.22 and 3.23 and the corresponding variance values for those components are:

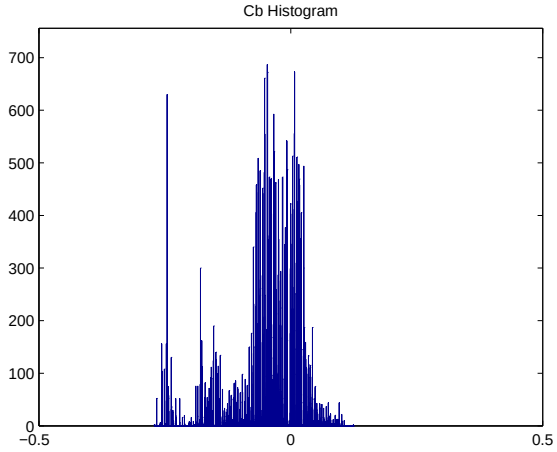
- $\sigma^2_{Cb} = 0.0035$
- $\sigma^2_{Cr} = 0.0011$

Accordingly, bell shaped Gauss curves are fitted to the data of the observations and illustrated in figure 3.11.

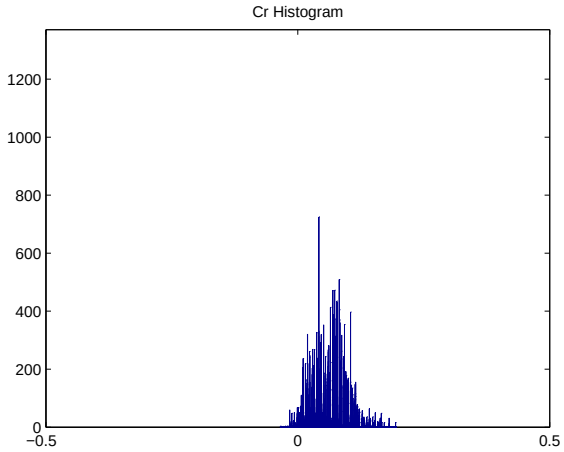
As mentioned in section 3.4 by calculating the covariance matrix of Cb and Cr distributions, joint pdf distribution of skin chrominance components can be estimated. Let us first illustrate the distribution of Cb and Cr components in $2D$ chrominance space which where held by training data (Figure 3.12. Finally by calculating the covariance between g and r and using



(a) Y Histogram

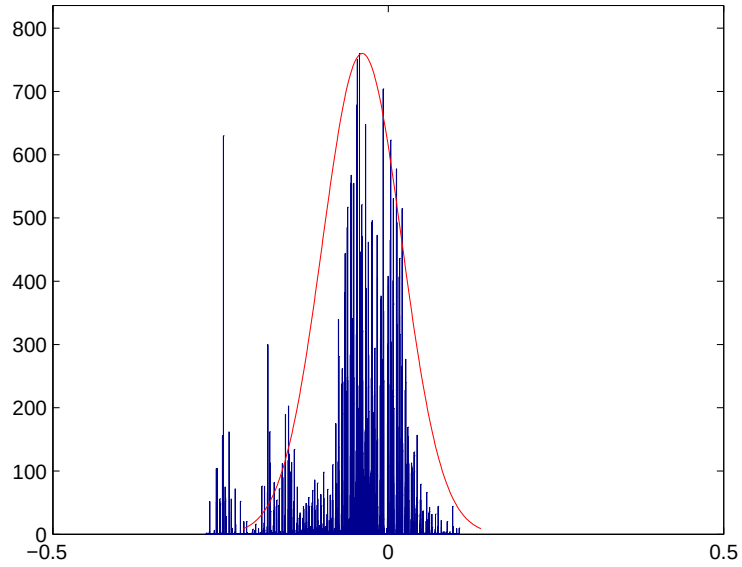


(b) Cb Histogram

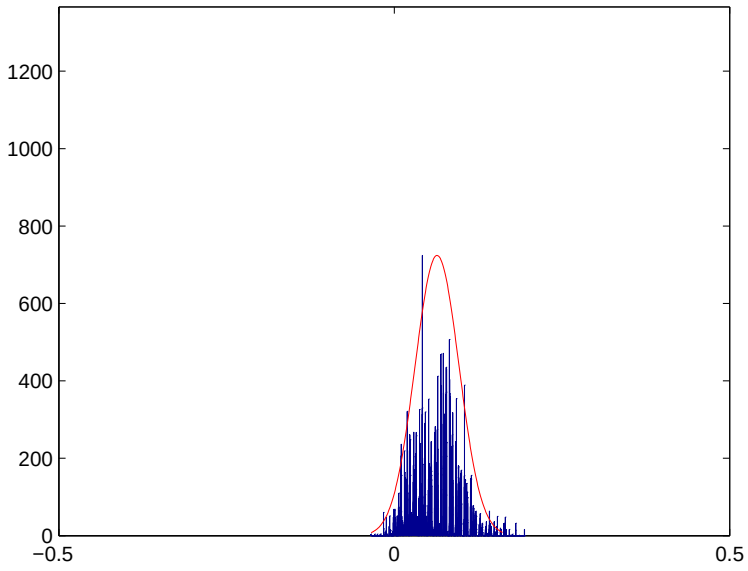


(c) Cr Histogram

Figure 3.10: Histograms of Y, Cb and Cr components for the skin pixels in training.



(a) Gauss distribution of Cb



(b) Gauss distribution of Cr

Figure 3.11: Gauss distributions of Cb and Cr components

the theoretical background mentioned in previous section the elliptical surfaces for the skin locus are extracted and illustrated in figure 3.13.

3.8 Comparison of the Color Space Performances for Skin Color Extraction

We have prepared a test bed to compare the skin color extraction performances under changing lightning conditions with different camera parameters and with different users. We have compared 2 color spaces, namely $n - RGB$ and $YCbcr$ color space performances. As mentioned in chapter 2, *n-RGB* and *YCbCr* color spaces are the most common two color spaces in skin detection analysis. An instance of our test bed is illustrated in figure 3.14).

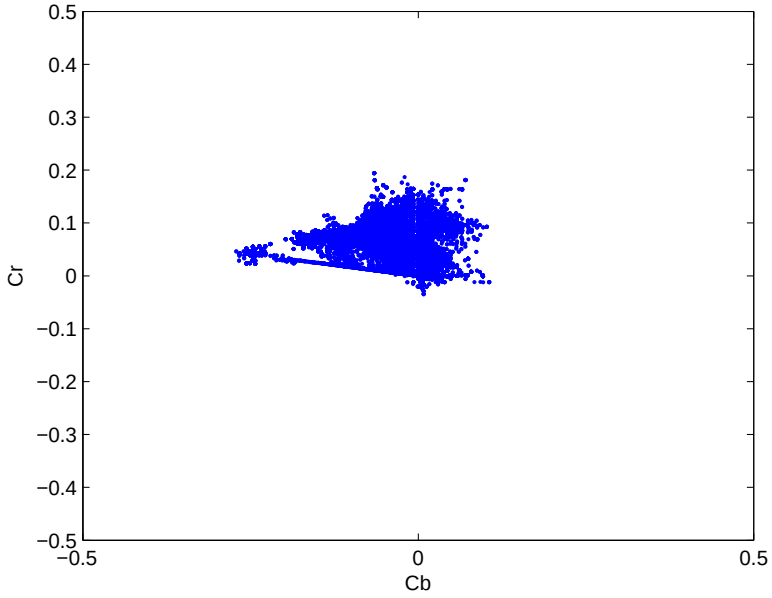
This test bed is constructed to compare the skin locus performances of *n-RGB* and *YCbCr* color spaces. The images to be analyzed can be selected from a recorded image, recorded video or it can be a snapshot of the currently acquired image by the camera. Training data are the sampled skin pixels that analyzed in the previous two sections. Those training data are compared to the test and the robustness of the skin loci of color spaces are analyzed for different conditions. To extract valuable information in the mathematical manner, Mahalanobis distances of the test data are calculated with respect to the training data. Mahalanobis distance is a useful way of determining similarity of an unknown data set to a known set and calculated by the following formula which was given also in section 3.4.

$$[\lambda_s(i,j)]^2 = [\ \underline{\mathbf{x}}(i,j) - m_s\ ]^T\ \mathbf{C}_s^{-1}\ [\ \underline{\mathbf{x}}(i,j) - m_s\ ] \tag{3.29}$$

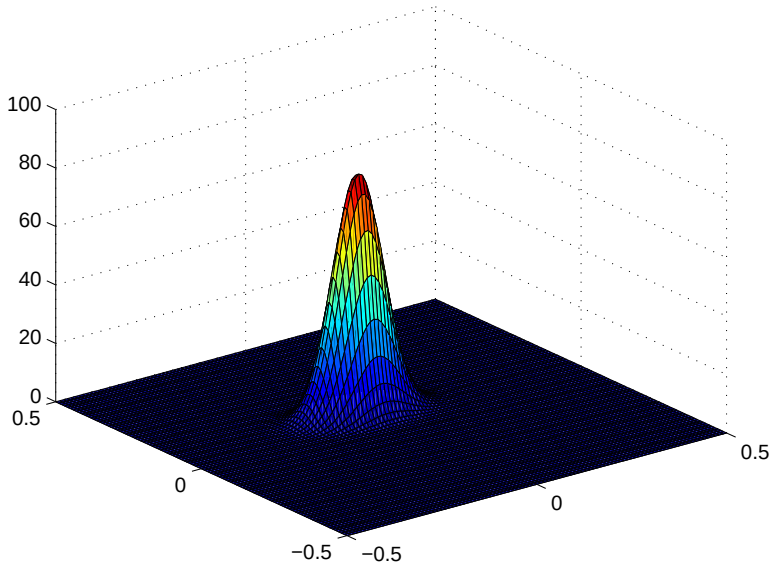
where

$$\underline{\mathbf{x}}(i,j) = \begin{pmatrix} r_(i,j) \\ g_(i,j) \end{pmatrix} \tag{3.30}$$

for *nRGB* color space and



(a) Distribution of $(Cb - Cr)$ in 2D



(b) Distribution of $(Cb - Cr)$ in 3D

Figure 3.12: $(Cb - Cr)$ distribution of skin pixels in training. Illustrates coarsely skin locus.

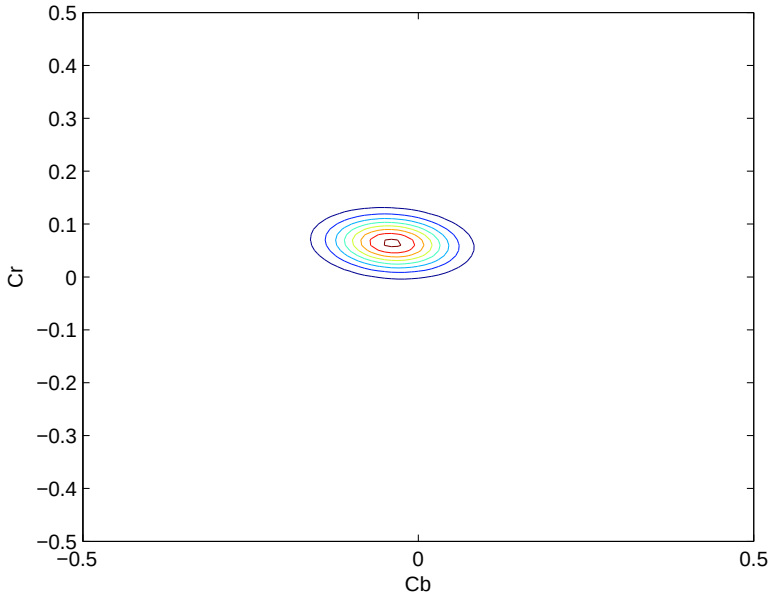


Figure 3.13: Skin Locus in ($Cb - Cr$) chromaticity diagram.

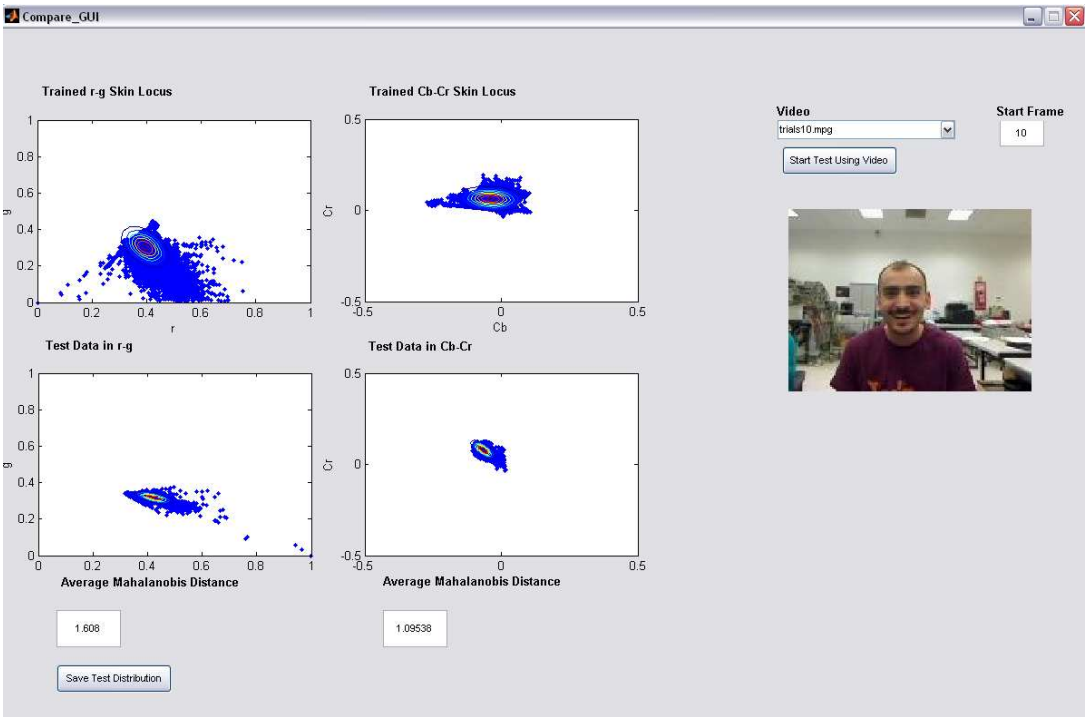


Figure 3.14: Skin Color Extraction performances of *n-RGB* and *YCbCr* Color Spaces

$$\underline{\mathbf{x}}(i,j) = \begin{pmatrix} Cb_(i,j) \\ Cr_(i,j) \end{pmatrix} \tag{3.31}$$

for *YCbCr* color space.

Mean values of *r*, *g*, *Cb* and *Cr* have already been extracted in previous two chapters and the corresponding covariance matrices $\mathbf{C}_{rg}$ and $\mathbf{C}_{CbCr}$ can be written as,

$$\mathbf{C}_{rg} = \begin{pmatrix} \sigma_r^2 & Cov_{rg} \\ Cov_{gr} & \sigma_g^2 \end{pmatrix} \tag{3.32}$$

$$\mathbf{C}_{CbCr} = \begin{pmatrix} \sigma_{Cb}^2 & Cov_{CbCr} \\ Cov_{CbCr} & \sigma_{Cr}^2 \end{pmatrix} \tag{3.33}$$

where covariance and variance values of each data can be estimated by the equations 3.21, 3.22 and 3.23.

Test results and comments of color space performances are in chapter 7.

CHAPTER 4

HAND SEGMENTATION

4.1 Introduction

Detection is one of the most challenging steps in many image processing systems. To analyze and extract valuable information from the acquired image, one needs to find the desired data in the entire set of pixels. Face detection is one of the most popular challenges among detection systems. By using face detection many valuable applications have been developed. Counting people in a room, recognizing an admin of a system, following a suspect in a street, etc. Many systems need to detect faces as a primary info. In the scope of this thesis, it aimed to detect the exact place of the hands in an image and then recognize the gesture performed by hands. Since the hand has not a strict shape like face, hand gesture recognition has less spot then face detection and recognition in literature. Most systems avoid handling hand systems because of this fact. This was one of the reasons why this thesis study was performed. To detect the hand(s) in the image a two steps system was designed. First, skin color locus for the current is extracted for the user's skin color, lightning condition and camera parameters. Then as the second step, hand is detected by eliminating false positive skin pixels and identifying hand(s) and other real skin color regions. The hand detection system in this thesis study was based on the method proposed by [2]. Also many features were added to the proposed method in [2] and all those details will be mentioned in the following sections of this chapter. Scope of the chapter is outlined as follows:

- Overview of Proposed Hand Detection Method
- Coarse Skin Color Extraction in $n - RGB$ Color Space
- Fine Skin Color Extraction

  - Extraction of $g_{pos} = (g - r)$ and $g_{neg} = (g + r)$ Histograms
  - Extraction of Fine Skin Boundaries
  - Frontal Face Detection to Decide the Fine Skin Color
- Hand Detection Using the Fine Skin Color Information
- Varying the Adaptive Thresholds for Changing in Lightning Conditions

4.2 Overview of Proposed Hand Detection Method

As mentioned in chapter 3, the skin color segmentation is very sensitive camera parameters and non-linearity in camera acquisitions (like extreme dark or bright pixels). So, as in many systems, the most critical part of the hand detection is having the right thresholds for the current conditions. The color of human skin can be studied under "global skin-color cloud" in some certain conditions [3]. In global skin color, color of the human skin is described in a more general way by taking the tolerance values large enough. On the other hand skin-color of a specified human under certain illumination should be described by more specific mean values and narrower tolerance values not to convey skin like pixels in consideration. In order to achieve the best segmentation, the optimal parameters should be chosen. To find the optimal values of the parameters, thresholds can be set manually at the beginning of the segmentation but this method would be time consuming and would ruin the user friendly environment of the application. Here a self calibration algorithm based on a histogram clustering method is used to determine the necessary parameters.

Normalized-RGB is the color space used to extract the skin color in this study. Thus, r (normalized Red in $n - RGB$ color space) and g (normalized Green in $n - RGB$ color space) thresholds must be initialized. System starts with a quick coarse skin color extraction as mentioned in [2]. This is done with using fixed thresholds and those thresholds covers a wide range to have all type of skin colors under many of the lightning conditions. Since the range of boundaries are so wide, the resultant image would have many false positive type errors. Achieved pixels are called skin pixel candidates and those pixels would go into a fine skin color extraction process to narrow the thresholds and extract true skin pixels. In fine skin color extraction, skin candidate pixels' $(g - r)$ and $(g + r)$ (namely g_{pos} and g_{neg}) histograms are extracted. Those histograms would have typically a few local peaks and each

one would correspond to a group of pixels. For instance, one peak corresponds to skin-like pixels (wood or similar) and the other peak would correspond to real skin pixels. Those thresholds construct the new narrowed boundaries for fine skin segmentation.s New narrowed boundaries are applied to skin candidate pixels and a frontal face image would be searched in all trials. If a frontal face image can be captured in a trial, then the corresponding thresholds extracted from g_{pos} and g_{neg} histograms would be the fine skin color thresholds for the current conditions. Once the fine skin colors are extracted, hand(s) will be detected using shape analysis. Skin pixels would be grouped by a region growing algorithm and it is assumed that just face and hand(s) would be the grouped pixels. By shape analysis (since the face has a strict shape), face will be found and the remaining group of pixels would compose hand information. All this process would have many internal variables like size of hand and face, local peak thresholds etc. Those variables were also designed in an adaptive manner and the details of the entire system will be mentioned in the following sections.

4.3 Coarse Skin Color Extraction Using $n - RGB$ Color Space

The idea behind the coarse skin color extraction is that eliminating the pixels which definitely do not belong to a skin region in the image in a quick way. This reduction in the pixels to be investigated would fasten the process in the following steps. Also for some cases just coarse skin color extraction would be sufficient to extract skin region in the image. Since coarse skin color extraction would not yield efficient results for most of the time, coarse extraction should be done quickly otherwise it would be computationally expensive for such a step. So using fixed skin color thresholds on the acquired image would be a good choice for this aim. One has to consider in coarse skin color extraction not to eliminate any real skin pixels, which means system could welcome false positive results instead of false negative results. Accordingly, skin color thresholds would have a wide range on the color space map to count the pixels as skin region for all types of skin colors (European, Asian, African etc.), all type of lightning conditions (bright, dark, daylight, fluorescent light, tungsten light) and for camera parameters. By considering these facts, many researchers proposed numerous skin locus distributions for $n-RGB$ color space [4]. Their results are consistent but most of them are ignoring the change in the spectrum of the light source (from blue to yellow). For that reason we have used an

adaptation of the skin locus proposed in [3]. Coarse skin region is bounded by:

$$g \leq r \tag{4.1}$$

$$g \geq r - 0.4 \tag{4.2}$$

$$g \geq -r + 0.6 \tag{4.3}$$

$$g \leq 0.4 \tag{4.4}$$

in $n - RGB$ color space and the illustration of the skin locus in $r - g$ domain is shown figure 4.1).

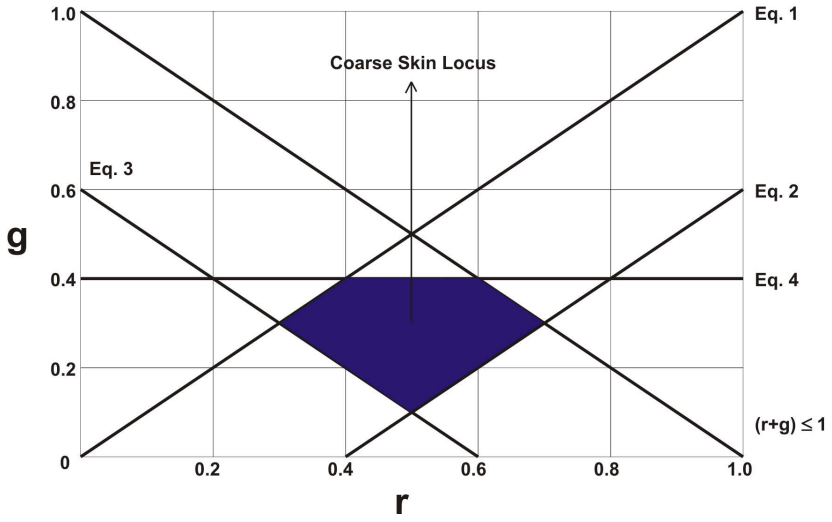


Figure 4.1: Skin Locus in r-g domain

$r + g \leq 1$ is the additional requirement comes from the normalization property of $n - RGB$ color space. As an adaptation to the [2], we have added another criterion namely 'Adaptive Bright Rejection Criteria' to the skin locus.

$$\frac{(R + G + B)}{3} \leq (230 - 255) \tag{4.5}$$

This criteria was added to eliminate bright pixels in the image. As investigated in chapter 3, pixels with minimum intensity ($I_{i,j} \leq I_{max}$) should be used because chromaticity calculations are unreliable due to the high level of noise in low RGB camera responses. Even for high RGB camera responses, the color information is distorted if one or some of the elements of a

pixel are under extreme illumination. For this reason it is assumed that each channel should be less than 8 bits and pixels having at least one element is 8 bits (255) are ignored directly. Because of the fact that most of the cameras have a non-linear intensity response, at higher RGB outputs can be set to a value around 240 [7]. But not to lose any data, I_{max} value was leaved as an adaptive variable because skin candidate pixels might be in that area for certain conditions. For instance, $R = 255, G = 255, B = 255$ (not n-RGB), corresponds to white pixels and yields $r = 0.33, g = 0.33$. Those pixels cannot be eliminated by typical skin locus region in $r - g$ skin locus. In literature, white pixels are typically added to skin locus because in very bright scenes, some areas on skin regions can be shinny and look as white pixels in the acquired image. But, how much white can be considered as a skin candidate, is a significant question and analogously it was considered as a variable in our approach which is in the range showed in equation 4.5. It varies proportional to the total brightness of the entire frame and it is set in the first few frames in video sequence by locating the face correctly in the images (method will be explained in following sections).

5 skin locus thresholds are applied to the input frame and skin pixel candidates are obtained. Figure 4.2 shows a simulation result of the coarse skin pixel extraction by the given 5 skin locus boundaries.

Figure 4.2: Coarse Skin Color Extraction Example

As clearly illustrated in figure 4.2, skin color candidates involve many false positive results. For instance, since wood color is a skin like color it is included in coarse skin color segmentation. It is an inevitable result because the system is open to all type of skin colors under all environmental conditions. So we need to eliminate those faulty skin pixels by deeper analysis.

4.4 Fine Skin Color Extraction

Once the coarse skin color extraction is applied to the input image, skin candidate pixels come up and a fine skin color extraction procedure is performed on those pixels to decide which ones are the real skin pixels. The aim of fine skin color extraction is grouping the skin candidate pixels by considering their resemblance to each other. This grouping of skin candidate pixels is performed by fine skin color extraction procedure presented in the following subsections.

4.4.1 Extraction of g_{pos} and g_{neg} Histograms

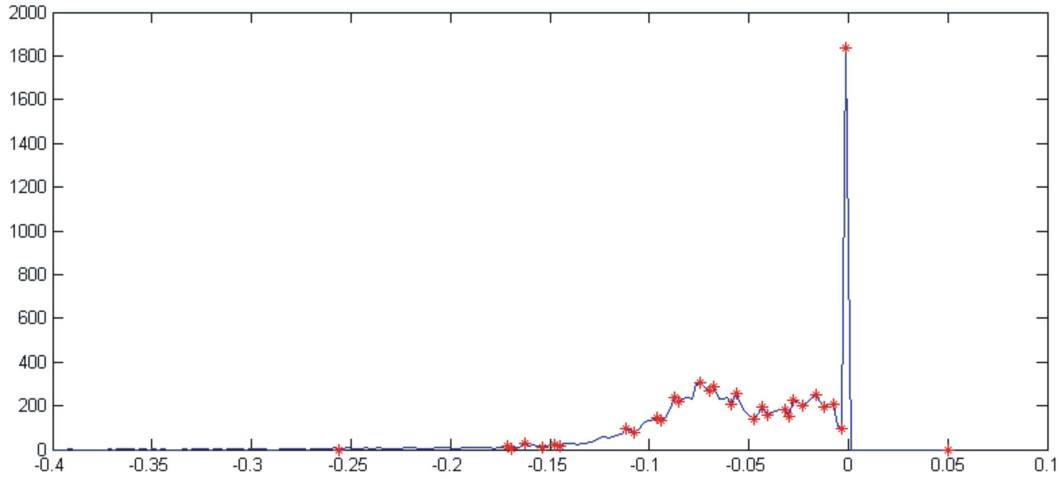
Fine skin color extraction is started by extraction of the $(g + r)$ and $(g - r)$ histograms of the skin candidate pixels obtained in coarse skin color extraction process. $(g + r)$ and $(g - r)$ histograms are chosen because the first 4 equations in coarse skin color extraction use these two variables. The corresponding histograms of the candidate skin pixels of the image in figure 4.2 are illustrated in figure 4.3.

Here, we need to make a decision if further processing is necessary or not. To make this decision, first a Kernel smoothing is applied on the histograms to reduce small fluctuations. Those fluctuations could be caused by the camera noise or natural skin color distribution could yield such a situation. In both cases, those few pixels might construct local peaks in the histogram and yield misleading results. For this reason such roughnesses in histograms are smoothed by nearest neighbor Kernel smoother where the smoothed results are the weighted averages of the histogram values on a sliding window. Kernel smoothing is defined as,

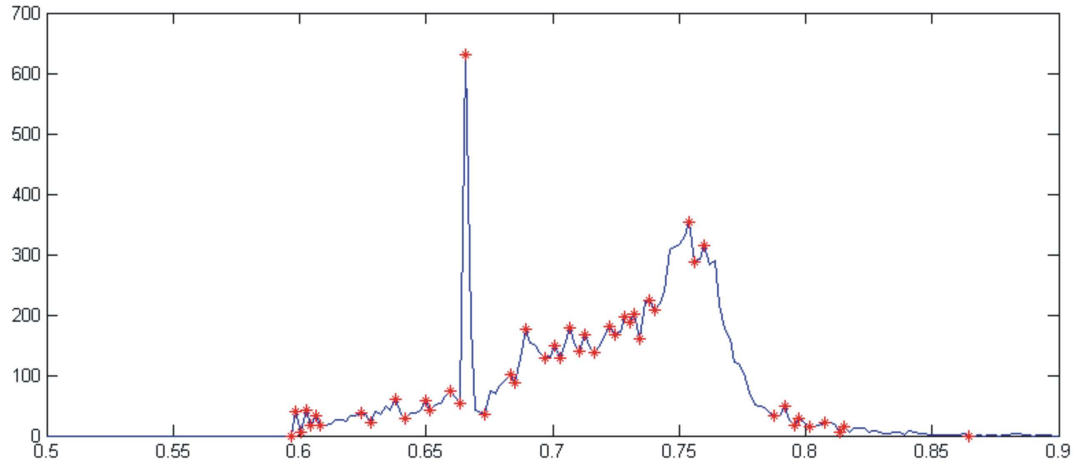
$$Y_x = \frac{1}{2n+1} \sum_{i=-n}^{n} [G(x+i)Y(x+i)] \tag{4.6}$$

where $2n+1$ is the window size and is proportional to the histogram size and G is the weighing factor which is a one dimensional Gaussian distribution calculated by,

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x)^2/2\sigma^2} \tag{4.7}$$



(a) g_{pos} $(g - r)$ histogram



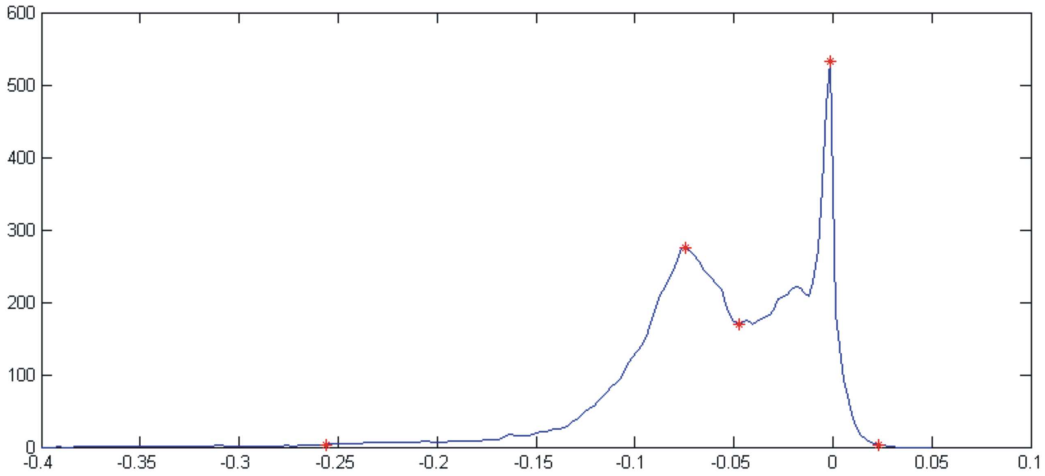
(b) g_{neg} $(g + r)$ histogram

Figure 4.3: Histograms for g_{pos} and g_{neg} for the candidate skin pixels in figure 4.2

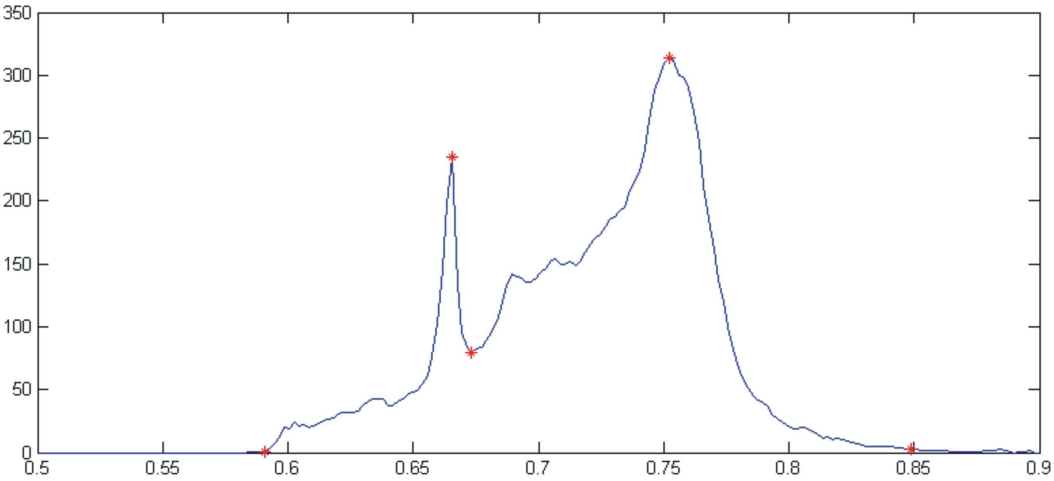
A typical Gaussian weighing window is:

$$[.006 \quad .061 \quad .242 \quad .383 \quad .242 \quad .061 \quad .006] \tag{4.8}$$

The smoothed version of the histograms showed in figure 4.3 which were extracted using equations 4.6 and 4.7 are illustrated in figure 4.4.



(a) Smoothed g_{pos} $(g - r)$ histogram



(b) Smoothed g_{neg} $(g + r)$ histogram

Figure 4.4: Kernel Smoothed Histograms and local peaks and valleys for g_{pos} and g_{neg}

Once the histograms are smoothed, data extracted from histograms become reliable and one can analyze further for fine skin color extraction. This extraction is performed to narrow the skin locus considering the skin candidate pixels. If we can estimate that fine skin color

extraction will not yield any extra information for the skin segmentation, then there is no need to investigate the input frame further. To decide further investigation is useful or not we use an entropy calculation, which was introduced in [5]. By finding the local peaks of the histograms, one can estimate the entropy of the candidate skin pixels. The following is the equation for entropy estimation.

$$ENT = \sum_{i=0}^{N-1} h(i) - Max[h(i)] \tag{4.9}$$

In equation 4.9, $h(i)$ corresponds to the histogram value for g_{pos} or g_{neg} distributions which were illustrated in figure 4.4. $Max[h(i)]$ is the global peak of the current histogram. ENT is the entropy value of each histogram and reversely proportional to the similarity among skin candidate pixels. Low entropy value corresponds to an image composed of entirely real skin or entirely a similar color to skin, like wood. On the other hand, a high entropy value means that skin candidate pixels have a wider distribution in the coarse skin locus. One can interpret such an outcome that there is a real skin area in the image along with some areas similar to skin color. Such a distribution is worth to investigate further and as a result fine skin color extraction is just applied for high entropy frames. Having one type of skin like colors in skin color candidates does not need any further investigation and fine skin color extraction step is skipped. If the entropy for skin candidate pixels is high enough, local peaks and valleys of the histograms are determined for further analysis. The smoothed and peak-valley extracted output for the input frame in figure 4.2 is illustrated in figure 4.4.

4.4.2 Extraction of Fine Skin Boundaries

The aim of fine skin color extraction is grouping the skin candidate pixels and deciding which group of pixels correspond to true skin color. In the previous section, by extracting the local peaks and local valleys of the g_{pos} and g_{neg} histograms, borders of those groups were formed. Each peak with its valleys compose a new narrowed skin color boundary. In figure 4.4, both g_{neg} and g_{pos} histograms have two peaks and each peak has two valley points around it. In fact, the peaks correspond to the skin and wood in the input frame. Since wood and skin are both fall in skin locus area, they can not be eliminated in coarse skin color extraction. Here, the boundaries which will divide the skin locus into smaller skin loci, are constructed by the

valleys of the histograms. Valleys compose a new threshold regions as follows:

Extracted valleys of g_{neg} histogram compose:

- Region1: From 1st valley to the 2nd valley
- Region2: From 2nd valley to the 3rd valley

Also two distinct regions are extracted from g_{pos} histogram in the same way.

- Region3: From 1st valley to the 2nd valley
- Region4: From 2nd valley to the 3rd valley

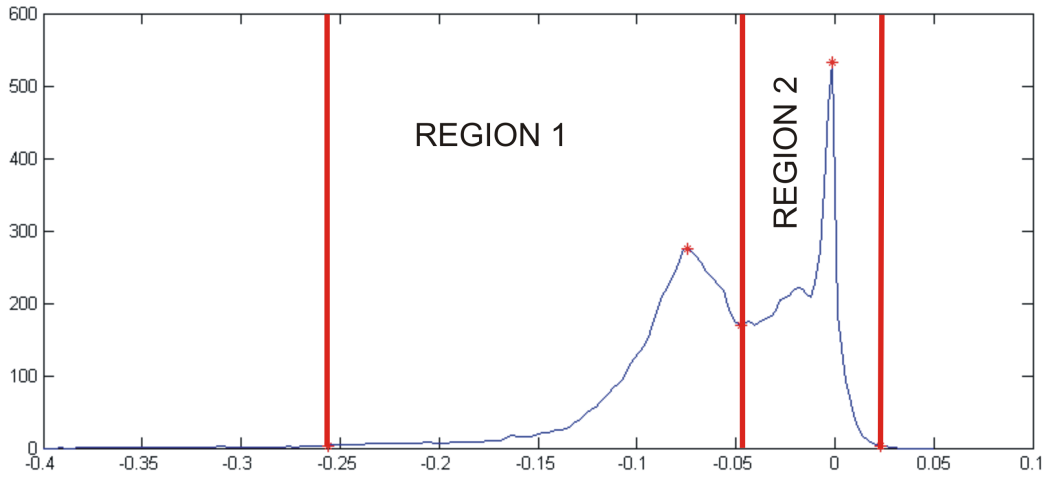
Illustrations of new boundaries are in figure 4.5.

Up to now, 4 new skin boundaries are extracted. By combining these 4 thresholds, *Skin Color Cloud* can be partitioned into new narrowed skin locus candidates. New skin locus candidates can be grouped by using the boundary regions shown in 4.5 and *New Skin Color Boundary* values are constructed by the combinations given in table 4.1.

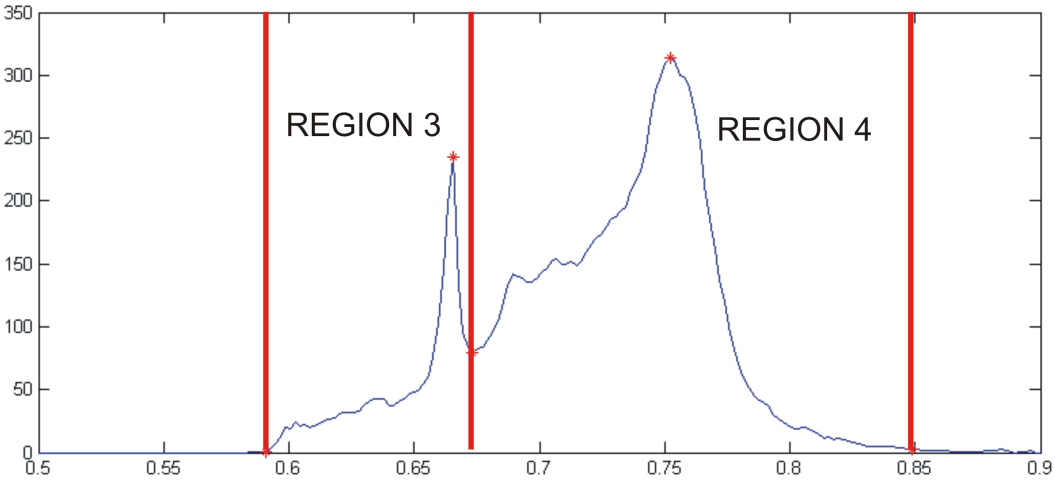
Table 4.1: New Skin Color Boundaries for Fine Skin Segmentation

NSCB1 : Region1 + Region3	NSCB5 : Region1 + Region3 + Region4
NSCB2 : Region1 + Region4	NSCB6 : Region2 + Region3 + Region4
NSCB3 : Region2 + Region3	NSCB7 : Region3 + Region1 + Region2
NSCB4 : Region2 + Region4	NSCB8 : Region4 + Region1 + Region2

Here we divided skin locus into 8 new smaller skin loci and correspondingly we have 8 new skin color boundaries. At this point we are sure that, at least one of the new skin loci will extract real skin in the image. Those new 8 boundaries are applied to the image composed of skin color candidate pixels and 8 images are formed where each one is constructed by applying a different $NSCB$. Those 8 images extracted from the candidate pixels given in figure 4.2 are illustrated in figure 4.6.



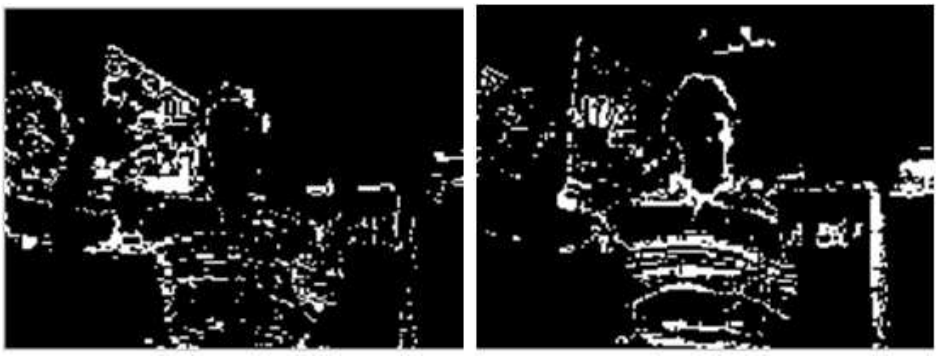
(a)



(b)

Figure 4.5: Smoothed Histograms and local peaks and valleys for g_{pos} and g_{neg}

(a) NSCB1 (b) NSCB2



(c) NSCB3 (d) NSCB4

(e) NSCB5 (f) NSCB6

(g) NSCB7 (h) NSCB8

Figure 4.6: Narrowed Skin Color Thresholds Applied to the Image in figure 4.2

4.4.3 Frontal Face Detection to Decide the Fine Skin Color

Frontal face detection is a well-known challenge in image processing world. In most of the systems, which involves human machine interaction, user's face is directed to the screen. Since, our system would be used in computers, frontal face detection might be a good choice to find the user's true skin color. At this point we have come up with 8 images, which was formed by using different narrowed skin color boundaries. Here all aim is to distinguish the right image which was compromised from real skin color. Once the right image is found corresponding thresholds composing that image would be finalized as the skin locus for the current conditions.

To detect the face, a fast algorithm searches all 8 images, if a frontal face is included in the image or not. When a proper face in the image is located, we can take out the corresponding $NSCB$ values as the final fine skin color segmentation thresholds for the recent lightning conditions, camera parameters and user specific environment. Some of the 8 candidate images correspond to real skin color, some of them correspond to non-skin color pixels and some of them correspond to the combination of them. To locate a frontal proper face in the images, we use a technique based on blob elimination. Blobs are the connected pixel groups in each bitwise image. For instance, hand in figure 4.6(b) is one of the blobs in all 8 images. The challenge of this section is to find out the distinct blob which belongs to frontal face among 8 images.

For this purpose we use a 3 steps method:

1. Height to width ratio of the blobs

In this step, all the blobs in all 8 images are searched. If none of the height to width ratios of the blobs in an image is proper to be a face then that image is eliminated. It is likely to have more than a few blobs are in the constraints of height to width ratio and needs further investigation.

2. Ellipse fitting

Images, not eliminated in the previous step, are subjected to an ellipse-fitting algorithm.

To fit ellipse around blobs a method based on the least square criterion is used. An ellipse can be defines by the following equation.

$$a_1 x_i^2 + a_2 x_i y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + f = 0 \tag{4.10}$$

Here all is need to estimate $A = [a_1 a_2 a_3 a_4 a_5]$ value by the least square estimator. The cost function for this aim is as follows,

$$e = \sum_{i=1}^{N} (a_1 x_i^2 + a_2 x_i y_i + a_3 y_i^2 + a_4 x_i + a_5 y_i + f)^2 \tag{4.11}$$

A value providing the minimum error in equation 4.11 is used to extract the orientation, major and minor axis information of the blobs. Determined ellipses are fitted to each blob and checks if the ellipse is fitted to the corresponding blob or not. For this reason, the following condition should be provided. 0.67 value was found empirically in the experiments which clearly distinguish faces from faulty blobs.

$$\frac{'Numberof Skin Pixels in Fitted Ellipse'}{'Numberof Pixels in Fitted Ellipse'} \geq 0.67 \tag{4.12}$$

If any of the blobs in an image is not a proper ellipse to be a face then that image is eliminated. After the eliminations if more than one image remains with blobs providing ellipse fitting criteria, then deeper analysis is needed to be done.

3. Facial Components Analysis

If there are still images more than 1, then blobs in those images are compared according to their facial components characteristics and the best match is signed as the face blob. For this reason location of eyes and lips are considered. Since the orientation and axis information of the blobs are extracted, face orientation is coarsely known and this information is used to locate two eyes and lips in the blobs. The blob assuring the best match compose the final thresholds for skin color extraction.

If only one image is left after first or second method, corresponding $NSCB$ values for the image are used as the fine skin color thresholds and the remaining steps are skipped.

In our experiments, height to width ratio have usually resolved the selection of the frontal face for simple background images. If the background is complex and there are skin like pixels in the background then it is likely to have have similar height width ratio with a proper

face. Those blobs can usually be eliminated by the second step, namely ellipse fitting. Very few of our experiments needed facial components analysis mentioned in step 3. For facial components analysis, two eyes and lips were selected as main facial components and their placement in the face was considered.

To have consistent fine skin color thresholds, we repeat this procedure in the cascading frames till the variance of $NSCB$ values are below a certain limit. Then the mean of the $NSCB$ values along with its variances are taken as the final thresholds and skin color segmentation is finalized.

4.5 Hand Detection Using The Fine Skin Color Information

Once the consistent fine skin color thresholds have been held, those thresholds are applied in the upcoming acquired images (if it is a video sequence or a live stream). A frontal face is assumed to be in the image and one or two groups of connected pixels are searched to attain hands and no analysis are done on those blobs since reliable thresholds were extracted in the previous frames. At this procedure, there are some limitations to avoid noise. If there is only one group of skin pixels, it is assumed that this is a face. If more than 1 group of pixels remain after application of the thresholds it is assumed that there is 1 frontal face and 1 or 2 hand(s) is(are) included in the image. Those blobs are pointed out as the detected hands.

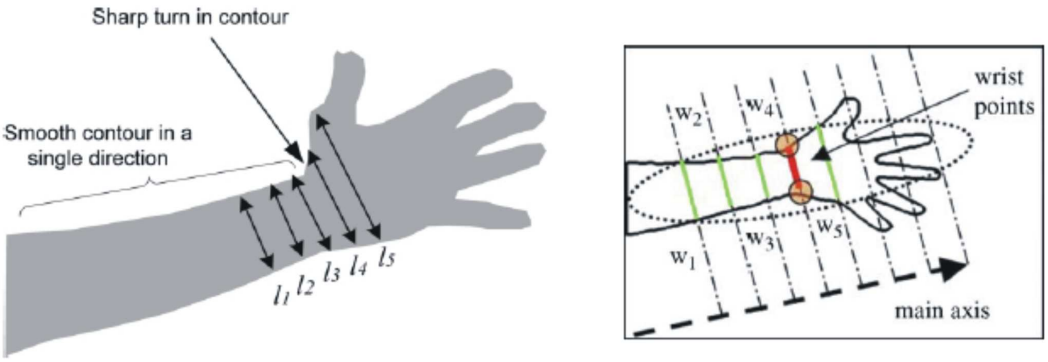


Figure 4.7: Two methods of wrist cropping [44][45].

However user might wear short or long sleeved clothes and if the user is wearing short-sleeved the arm and hand should be separated. In literature this procedure is called wrist cropping

and basically there are two methods for this purpose as illustrated in figure 4.7. In the first technique [44], change in the orientation of the arm at wrist-hand intersection line is pointed out as the cropping spot. Detected edge in the arm would have a smooth orientation till the wrist and at the wrist-hand intersection line the orientation is expected to change. However this method is not so robust to noises. If the arm is not segmented clearly the orientation on the arm edge will have fluctuations and this will yield misleading information. In the second method [45], blob itself is used instead of just edge information. Typically the hand will be thicker than the wrist-hand intersection line and also arm is getting thinner from elbow to wrist. So when the arm is traversed from elbow to hand the point where arm is getting thicker rather than getting thinner is the line for wrist cropping. Noise also might yield misleading information for this method but in our experiments, we have detected that this method is more robust to fluctuations in the segmented arm. Experimental results for the comparison of two methods are in chapter 6.

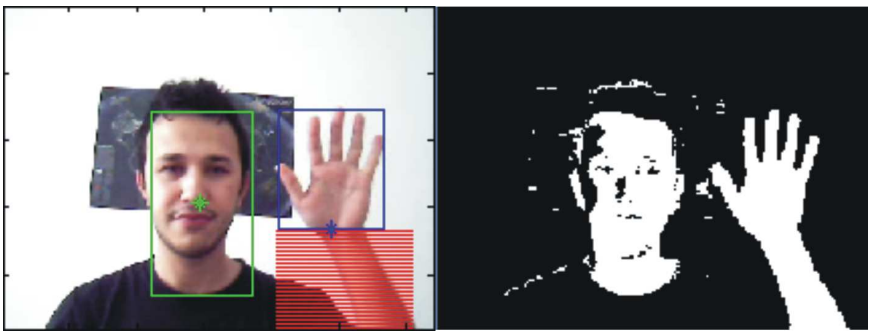


Figure 4.8: Illustration of wrist cropping in an experiment.

An illustration of the discussed hand segmentation procedure with wrist cropping is shown in figure 4.8. Input image is in the left and the fine skin color segmented image is in the right. As clearly seen, wrist is cropped at the hand-wrist intersection line and the hand is segmented clearly. Face is not considered if it is detected perfectly or not because the scope of this thesis study is limited with hand segmentation and hand gesture recognition. The occlusion case of hand and face is also omitted in this study. Face and hand is expected to be in distant places and they should not be in the same vertical alignment which means hand can not be in just above or below the face. Test results for the hand segmentation will be declared in chapter 6 and those results will be discussed in chapter 7.

CHAPTER 5

HAND GESTURE RECOGNITION

5.1 Introduction

Hand gesture recognition is an improving topic which has many applications areas in human computer interaction systems. It is believed that future systems will have so many frameworks however applications will be less useful. In modern systems design and interaction are crucial keywords to attract consumer attention and gesture recognition is bringing an advanced interaction environment in many systems. People would want to control sound or stop/play features with hand gestures while they are watching movies on their home theater systems or many people do not want to touch screens in ATMs while they are drawing money or some disable people can not use mouse or keyboards but they can still have some defined gestures for themselves to control computers. As you see in these examples, there are millions of application areas of hand gesture systems. It is very critical that those systems would work in environment free conditions and they would work in an efficient way to recognize gestures. Today's systems have many limitations especially for environmental conditions. We have proposed new methods to decrease the dependency to lightning conditions, camera parameters or user specific environment to detect hands in chapter 4. In this chapter, detected hand will be searched to recognize the current gesture. We have also proposed a new method to increase the recognition rate of a visual based recognition system.

Up to now, true skin color boundaries were held and using those thresholds and applying shape analysis hand(s) in the image was(were) detected. We have already located the face and hand as mentioned in chapter 4. Here we just need to apply the thresholds of the current condition's skin locus and detect hands by those shape analysis procedures for the upcoming

frames. Once the hand is located in the acquired image, rest is regarding the gesture recognition procedure. Hands are searched by a special profile extraction technique and gesture is estimated. The proposed method for gesture recognition in this study is based on a procedure called centroidal profile extraction which was mentioned in [5] and [7]. According to the centroidal profile extraction method, growing circles are drawn around the palm of the hand. Each circle is considered as a contour to move on and a polar transformation is used to count the number of fingers being shown. If a point on a circle is a skin pixel then the corresponding angle on the *Number of Skin Pixels vs. Angle* graphic will be increased by one. Skin pixels vs Angle histogram of this growing circles are extracted and peaks on that histogram are counted. Number of peaks in this histogram will give the number of fingers being shown to the camera. The histogram is extracted by using the following equation 5.1.

$$A(\theta) = \sum_{r=R_{min}}^{r=R_{max}} I(r, \theta) \tag{5.1}$$

where R_{min} is the radius of the smallest circle and R_{max} is the radius of the biggest circle to be drawn around hand. R_{min} and R_{max} values will be determined proportional to the size of the hand. An instance of the centroidal profile extraction method is in figure 5.1.

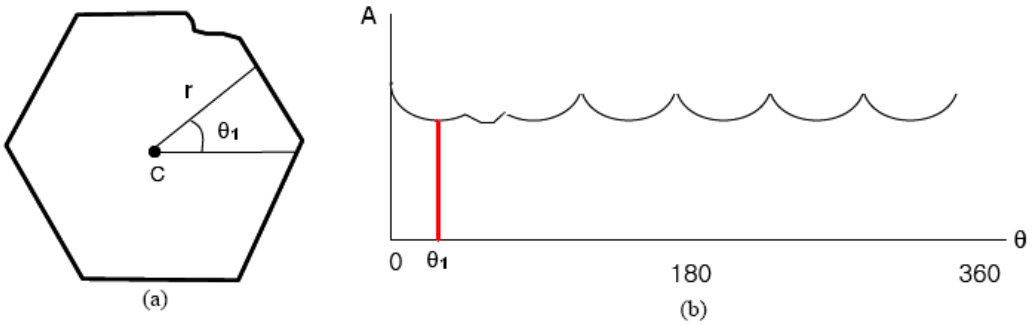


Figure 5.1: Instance of Centroidal Profile Extraction [5]

In the scope of this thesis, hands are assumed to be upwards (as a typical PC user would do). So just the $0 - 180\,^{\circ}$ or $180 - 360\,^{\circ}$ intervals are considered according to the starting point and direction of the circle contour. In figure 5.1, the contour starts from the right mid point of the shape and moves in counter clockwise direction. By the way, all the pixel values in the shape are assumed to be skin pixels but just the outer contour of the shape is drawn for visualization

purposes. Centroidal profile extraction performed on a segmented hand is visualized in figure 5.2

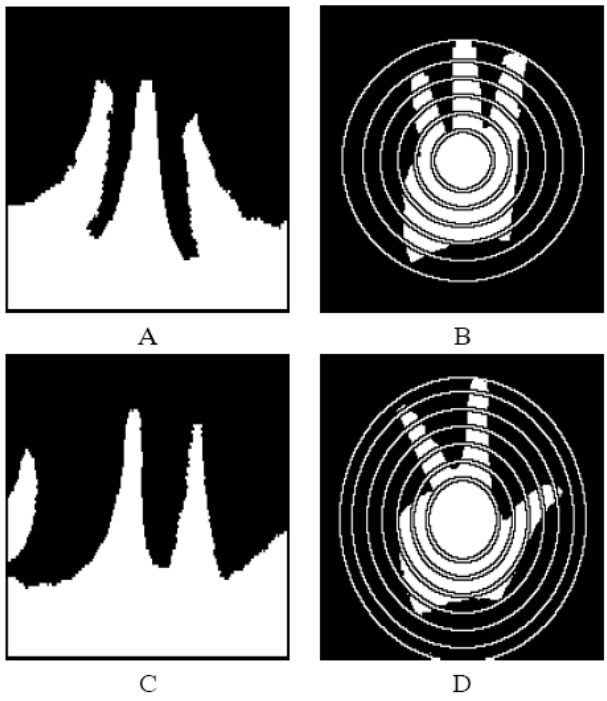


Figure 5.2: Polar Transformation Results of Two Hand Instances [7]

Our proposed profile extraction technique is adding some new features to the mentioned algorithm.

5.2 Hand Anatomy and Defined Gestures

Hands are the chief organs to interact with the environment physically. They have the best positioning capability and with this speciality they are the center for sense of touch. Human hand have 27 bones: 8 of them are in the wrist, palm contains 5 and fingers have remaining 14 bones as illustrated in figure 5.3.

Since hand gestures are based on the forms of fingers our defined gestures were chosen on the different orientations of the fingers. Most applications such as home theaters or games would just need 5 or 6 different inputs and their combinations. Therefore, our defined gestures are

Figure 5.3: Hand skeleton

the number of fingers being shown to the camera namely 1,2,3,4,5 and a punch which have no fingers shown. You can see some of our defined gestures in figure 5.4

5.3 Typical Hand Gesture Profile Extraction

Profile extraction of hand gesture typically finds the center of the palm and counts the skin pixels around that center. The center of the hand is usually located by finding center of mass of the bounding box, which is lying around the detected hand. Once the center of the palm is pointed, a small circle will be drawn around that point. The pixels on the circle are searched in the sense that if the pixels are skin color pixels or not. If all the pixels are skin color pixels, we can infer that our circle is totally lying inside the hand and then another circle is drawn with a bigger radius and the pixels on the new circle are searched also. It goes on till the circle crosses the pixels that are composed of the fingers. Once the fingers are achieved some pixels on the circle would be skin color pixels and some would not. Then just the skin pixels' angles are recorded for that circle to the profile extraction histogram. A bigger circle will be drawn again. Pixels on this new circle will also be searched and the angles of the skin color pixels will be recorded. This procedure will end when the circle has all non-skin color pixels, which means the circle is bigger than the hand. Once this circle drawing issue is finished the (skin pixels-angle) histogram will give an idea about the profile of the fingers. By considering the (pixel count vs. rotation angle) distribution one can estimate the hand gestures such as the number of fingers being shown or recognize a punch. Here, we count the local peaks of

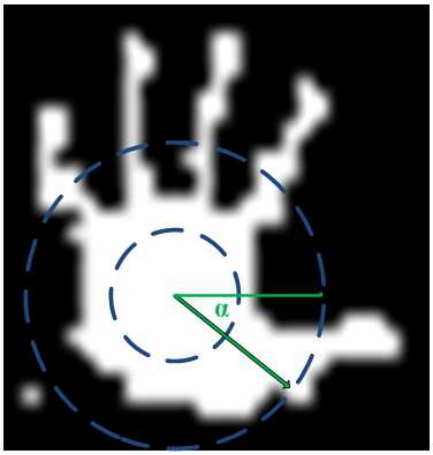
(a) Gesture 1 (b) Gesture 2

(c) Gesture 3 (d) Gesture 4

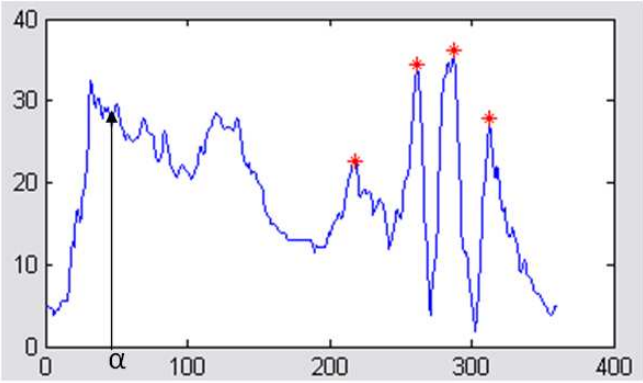
(e) Gesture 5 (f) Combination of Two Gestures

Figure 5.4: Some of the Defined Gesture Instances

the histogram and if the peaks are above a threshold value then we conclude that a finger is being shown in the image. Peak count would give the number of fingers being shown. An instance of input hand profile and its corresponding rotation angles vs pixel count histograms is illustrated in figure 5.5.



(a) Hand Gesture with Surrounding Circle Around it



(b) Skin Pixel Count vs. Rotation Angle

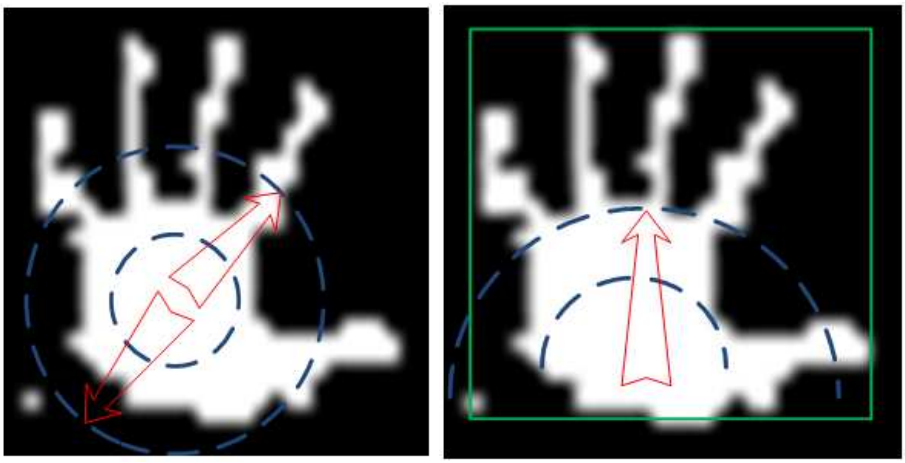
Figure 5.5: An example of Typical Gesture Extraction (Hand is the zoomed version of the detected hand in 4.2).

As clearly seen in figure 5.5, typical hand gesture profile extraction method has some disadvantages and leads misleading results. Here the thumb could not be found correctly in the pixel count vs. rotation angle histogram. In this type of gesture profile extraction surrounding circles are centered in the center of palm. This is inconsistent with the nature of the hand.

5.4 Proposed Hand Gesture Recognition Method

In the typical profile extraction method, skin pixels around the palm are counted by the method mentioned in section 5.3 and the orientation of those skin pixels are considered. In our proposed method we consider the nature of the hand skeleton. Assigning the center of the palm as the center of rotation is yielding faulty results because fingers are joining in the hand-intersection line (see figure 5.3) and the pixel count process must be initiated from this point.

In our proposed method we try to estimate the orientation of the hand at first. For this reason, we find the hand-wrist intersection line by contouring on the bounding box around the detected hand. If the right and left sides of the hand do not lie in a straight line, then the only continuous skin pixels on the bounding box lines will be the pixels corresponding to the hand-wrist intersection line as illustrated in the figure 5.6. If the hand-wrist intersection line can be found for a hand image, then the middle point of that line will be the starting point of the profile extraction technique (it was center of the palm in the typical profile extraction method). Also we draw just a half circle in the direction of hand center. The difference between the typical and proposed hand gesture profile extractions are illustrated in figure 5.6.



(a) Typical hand gesture profile extraction. Centered on palm.

(b) Proposed hand gesture profile extraction. Centered on hand-wrist intersection line.

Figure 5.6: Typical hand gesture profile extraction and our proposed method (Hand is the zoomed version of the detected hand in 4.2).

If the hand-wrist intersection line cannot be detected clearly, then the typical profile extraction is applied to the hand gesture.

The resultant histogram for the proposed profile extraction is illustrated in figure 5.7.

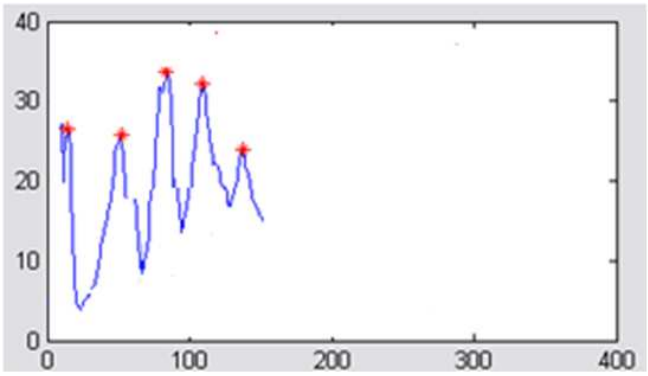
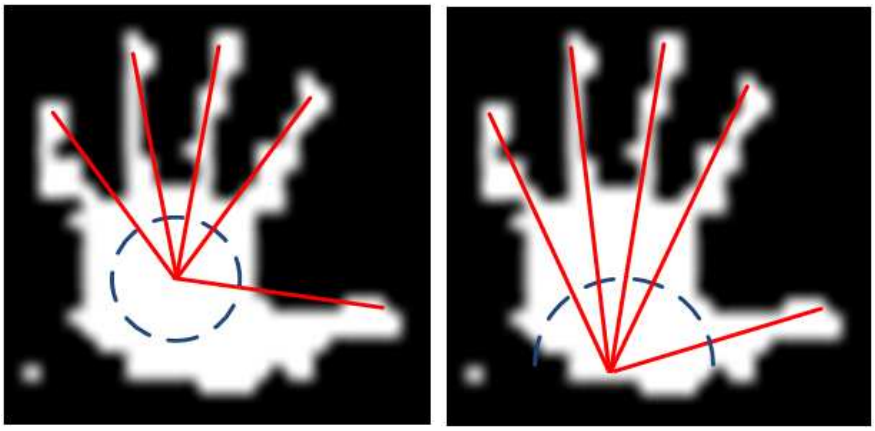


Figure 5.7: Extracted histogram of the proposed method for the same input image in figure 5.5

First merit of our method is that only half circles are drawn for hand gestures which decrease the computation time of the gesture recognition process significantly. Second and more serious advantage of our method comes out from changing the starting point of the profile extraction. When the center of circles is center of the palm, it is difficult to distinguish the fingers in the side, like thumb, because they do not lie in the direction of the center as illustrated in fingure 5.8(a). Clear illustration of two methods could be compared visually in figure 5.8.



(a) Direction of the skin pixel count when centered at palm

(b) Direction of the skin pixel count when centered at wrist-hand intersection line

Figure 5.8: Comparison of two methods. Typical profile extraction intersects non-skin pixels and yields misleading results.

In figure 5.8(a), thumb constitute a clear peak in the histogram for a certain rotation angle.

However our proposed method gives more accurate result since the center of the profile extraction is the joint spot of the fingers. Consequently, thumb will increase the histogram value for a certain rotation angle, there would be a clear local peak in the profile extraction histogram.

The key point of the recognition algorithm is determination of the local peaks. A point in the histogram is considered a maximum local peak if that point's value is preceded (to the left) by a lower value and the condition of the difference between those values should satisfy the following condition.

$$m_n - m_{n-1} \geq DELTA \tag{5.2}$$

$DELTA$ value is determined by the size of the histogram. It is directly proportional to the total number of pixels in the half circle surrounding the detected hand. However determining the right $DELTA$ value is so critical it does not guarantee to detect peaks in a perfect way. Hesitations may be occur due to the noise components around hands. For this reason such roughnesses in histograms are smoothed by nearest neighbor Kernel smoother where the smoothed results are the weighted averages of the histogram values on a sliding window. Kernel smoothing is defined as,

$$A(\theta) = \frac{1}{2n+1} \sum_{i=-n}^{n} [A(\theta + i)] \tag{5.3}$$

where $2n + 1$ is the window size and is proportional to the histogram size and A is the histogram vector held by the formula given in 5.1. This smoothing is necessary to deal with such fluctuations on the histogram. On the other hand, smoothing might yield loss of some valueable peak information. If two of the fingers are very near to each other and if the smoothing window size is bigger than the difference of the finger peaks then it is likely to lose one of the peaks. This trade off is critical in the design process. We have still having difficulties to determine the right peaks in the histogram. Clear examples and test results regarding this problem will be mentioned in chapter 6.

CHAPTER 6

TEST RESULTS & APPLICATIONS of THE THEORY

6.1 Introduction

Hand gesture recognition is a subtopic in machine vision and it mainly aims to decrease the effort in human machine interactions. Although there are serious improvements for input devices in computer world, people still find the interaction uncomfortable. Especially moving, rotating, scrolling, zooming and selecting can be quite exhausting in long studies if they are needed to be used frequently. While reading an e-book or looking a map on a computer screen, scrolling the page or moving the window will be quite easy by using just hand gestures. User can move a window by opening and moving his hand or select another window by showing the index finger on that window or scroll the page by showing a freezed thumb and a moving index finger. Not just typical computer interactions need such gestures, some game controlling features can be done by using static and dynamic hand gestures. Playing a boxing game with a controller free environment by just using hand gestures would be more realistic and enjoyable and also it could be a good exercise as a sport. Moreover, hand gestures might be a good replacement for touch-screens in public areas. Many people do not prefer to touch such buttons or touch screens (like in ATMs) for hygienic purposes. Hand gesture recognition technology does not aim to disable keyboard or mouse in full sense, but some applications, like the mentioned ones, would be much easier, comfortable and enjoyable by using the hand gestures. Our basic aim in this chapter will be finding effective solutions to these challenges by using the theory of this study. To check if a correctly working system can be constructed by the mentioned theory in this study, we need to measure the performance of our algorithm. For this reason we have tested the theory of the theory and obtain reasonable results.

6.2 Skin Color Modeling Tests in Different Color Spaces

To see the effect of users' skin color and camera parameters on the skin color segmentation process, we have prepared a test bed mentioned in chapter 3. Different lightning conditions, with different users and with different cameras were compared. By the formulation given in chapter 3, the test data are compared with the training data. Test images were chosen as; different user with same camera, same user with different camera and same user with same camera. For each case, the lightning conditions of input images were in a wide range. The instances of the test images are in in figure 6.1.

Figure 6.1: Test image samples. Skin pixels are sampled as test data.

In the experiments, it was observed that extreme lightning conditions always yielded non-linearities in acquired image pixel information. This dropped the performance of our algorithm especially when the extreme lightning conditions are not uniform on hand and face. If a sun light is directly coming into a room and lightning just one side of the user it would cause misleading skin color calibration. For this reason, we avoided to have non-uniform extreme lightning conditions in our experiments. If the lightning is not extremely shinny or dark, then the skin color could be calibrated correctly.

Table 6.1 gives the data for the Mahalanobis distances of color spaces. Smaller Mahalanobis distance results in better performance which means the corresponding color space keeps its skin locus for different conditions.

Table 6.1: Results of Skin Locus Comparison

Mahalanobis Distances for Test Samples of Skin Pixels		
	(r-g) Color Space	**YCbCr Color Space**
Same user as in training, Same Camera		
Similar Lightning	0.408	0.881
Lighter Lightning	0.886	1.190
Darker Lightning	1.708	4.751
Different User, Same Camera		
Similar Lightning	0.620	1.430
Lighter Lightning	1.057	3.740
Darker Lightning	2.416	4.218
Same user as in training, Different Camera		
Similar Lightning	4.347	2.827
Lighter Lightning	3.509	1.923
Darker Lightning	7.585	3.212

where some instances of the test images are shown in figure 6.1.

According to the results, obtained in table 6.1 both *nRGB* and *YCbCr* color space are robust against different skin colors. They maintain their skin chromaticity locus for different users, even for the same camera parameters user 1 has better results than the user participated in training. However for different lightning conditions *nRGB* color space maintained its skin locus better. *nRGB* is more robust for the spectrum of the lightning source changes like the difference between the daylight and fluorescent light. However, when it comes to the camera parameters *YCbCr* yielded an intense performance over *nRGB*. *YCbCr* had nearly two times smaller Mahalanobis distance values for different camera conditions. Also for very dark regions on face, (e.g. neck when the lightning is subjected from top) *nRGB* shows very poor robustness. We have observed this empirically by selecting the dark pixels on skin and calculating their Mahalanobis distances. This fact is caused by the non-linear transformation property of *nRGB* color space. *RGB* to *YCbCr* is a linear transformation and it is more robust for extreme luminance cases. *YCbCr* might be the choice when the linear chromaticity should be conveyed for different camera parameters. On the other side, if the application is specific to a distinct camera and if the conditions do not yield extreme cases using *nRGB* color space would be the right choice.

In addition table 6.2 clearly shows the *RGB-nRGB* and *RGB-YCbCrB* transformation durations for 1000 images by using matrix operations using MATLAB on a CoreDuo Laptop. If the input image is a (0-255) *RGB* image (as in most cases) using *nRGB* color space would yield a significant advantage on computation time. This small merit might be a critical choice reason in the manner of *nRGB* for real time applications. If the input image is a *nRGB* image then the choice would not effect the total computation time of the algorithm significantly.

Table 6.2: Computation time of Transformations

Transformation	Duration in seconds (for 1000 transformation)
RGB-nRGB	0.8408
RGB-YCbCr (when MATLAB function used)	3.9425
RGB-YCbCr (when Matrix operations used)	1.1463
RGB-nRGB (when RGB is norm-RGB [0-1])	0.7746
RGB-YCbCr (when RGB is norm-RGB [0-1])	0.7073

6.3 Tests of The Overall Gesture Recognition System

To measure the success rate of our system, a test bed was constructed and implemented in a Core Duo PC with the help of MATLAB. An interface was constructed to show the details of the algorithm. Hand detection and hand gesture recognition performance is measured in the test bed shown in figures 6.2 and 6.3. Figure 6.2 shows the hand segmentation process. In the left-top part, visualization of the process can be adjusted like enabling wrist cutting on the screen or enabling skin locus updates at each frame etc. In the left-middle the information of the hand segmentation can be observed like how many frames were processed, what are the computation times of sub algorithms etc. In the right-top part current frame and the skin candidate pixels can be seen. Skin candidate pixels are the pixels obtained by applying coarse skin color thresholds on the current frame as mentioned in chapter 4. Finally in right-bottom part there are two histograms with their pointed local peaks and valleys can be observed. They belong to the g_{pos} and g_{neg} histograms which were presented again in chapter 4.

Figure 6.3 shows the hand gesture recognition process. In the left-bottom part the success of the recorded video is shown. Each gesture information of each frame is recorded to a database

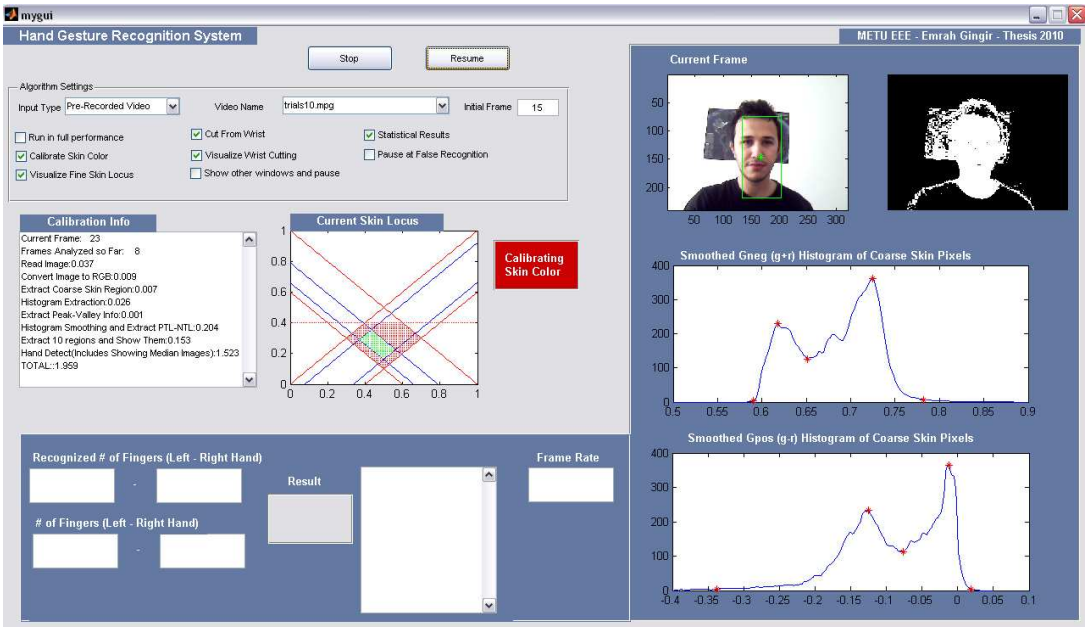


Figure 6.2: Instance of the main test bed in hand segmentation process.

and the current recognition result and the gesture in the database is compared if it is a correct recognition or not. If a correct recognition is achieved a green box appears otherwise a red box would light. Also the instant frame rate of the algorithm is shown in this part of the test bed. In the right-top part of the figure, input frame and the fine skin thresholds applied version is shown. Also face and hand are enclosed by colored boxes and wrist cutting procedure is visualized. Fine skin thresholds are obtained from the hand segmentation process and used throughout the application. Also this fine skin locus is illustrated in the middle of the test bed. Finally in the right-bottom part centroidal profiles of hands are visualized. Since there is only one hand in the current frame in figure 6.3, there is only one centroidal profile.

First our algorithm was compared with the previous studies' results. We have constructed a database of videos composed of different gestures by different users under different lightning conditions and with different camera parameters. However in literature most of the systems are either for special purposes or suffer from the lack of the desired features (e.g. not having color or very small size of the samples). Our system was designed for a typical PC user and his web-cam on the monitor. The traditional database are not suitable for our design. But we have implemented a variation of our algorithm to compare the recognition rate of our algorithm with previous studies. According to the implemented algorithm, hand segmentation

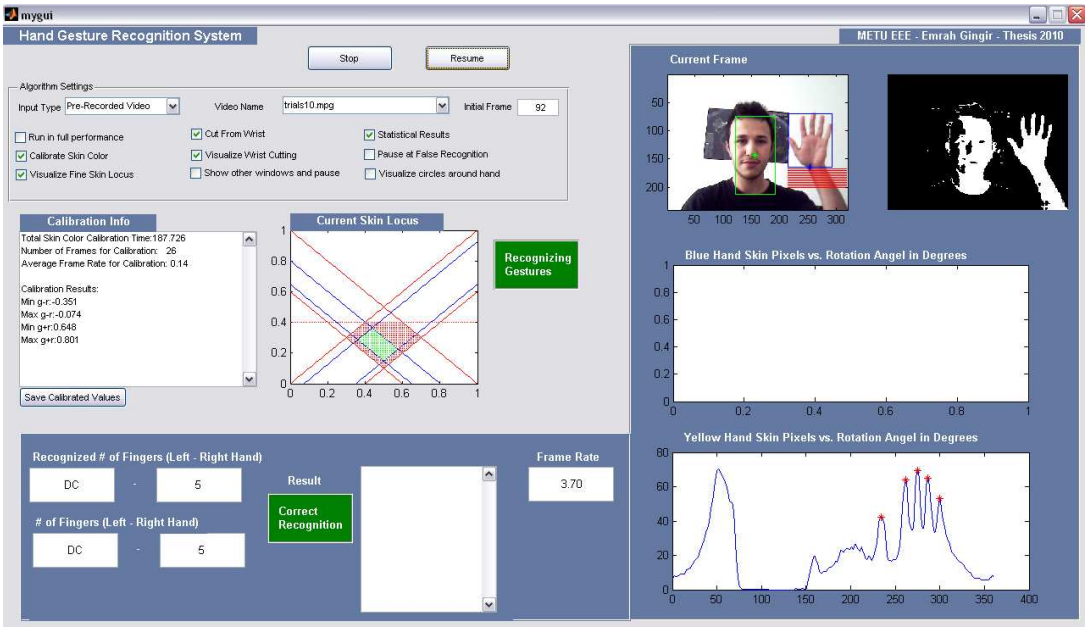


Figure 6.3: Instance of the main test bed in hand gesture recognition process.

part is not performed which was mentioned in chapter 4. Just the gesture recognition part of the algorithm could be compared with the previous studies since our system is an innovative one with user's whole body could be in the image with a complex background. The test images were selected from the *Cambridge Hand Gesture Data Set*.

As clearly indicated in table 6.3 our algorithm is much more open to realistic cases and has compatible recognition rate with the other algorithms which needs additional markers or training data. Previous studies have many constraints such as multi-colored gloves, blue screen, training etc. Our system was tested in a black screen test set since there is not a clear test set to perfectly compare two studies. Some instance images from the selected data set are illustrated in figure 6.4. Here just the lightning conditions and hand postures are changed for each gesture with disabling the background problem.

To entirely measure the performance of our algorithm we have constructed a test set composed of people sitting in front of a computer and camera is located on the screen. Test videos are chosen from the videos used to compare the skin color performances. Instances of those videos were shown in figure 6.1. Some of the videos are not challenging with simple backgrounds and with clear finger exhibitions. However some of the videos compel the success

Table 6.3: Comparison of Gesture Recognition Methods

Reference	Primary Method of Recognition	Number of Gestures Recognized	Background to Gesture Images	Additional Markers Required	Number of Training Images	Accuracy
39	Hidden Markov Models	97	General	Multi-colored gloves	400	91.7%
40	Hidden Markov Models	40	General	No	400	97.6%
41	Linear approximation to non-linear point distribution models	26	Blue Screen	No	7441	95%
42	Finite State machine modeling	7	Static	Markers on glove	10 sequences of 200 frames each	98%
43	Fast Template Matching	46	Static	Wrist band	100 examples per gesture	99.1%
This study	Feature Invariant	6	Black screen	No	No	98%

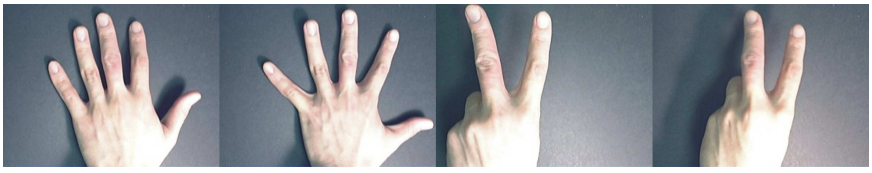


Figure 6.4: Instance images from the data set for comparison with previous studies.

rate of the system. Video1, Video2, Video 3 are easier to segment with simple backgrounds. Video 4 is recorded from distant and the user is wearing long sleeved. Video 5 has a complex background with non uniform poor illumination on skin parts of the user. The success rate of the videos are summarized in table 6.4.

Table 6.4: Recorded Video Test Results

Detailed Results	Video 1	Video 2	Video 3	Video 4	Video 5	Video6
Video Info						
Total Frames	1007	774	718	956	81	334
Used for Skin Color Calibration	13	15	13	15	15	14
Having Defined Gestures	818	591	632	684	60	265
Detection Results						
Correctly Recognized Frames	787	549	602	642	47	227
False Recognized Frames	31	42	30	42	13	38
Frame Rates						
Avg. FR for Recognition	8.08	6.64	8.59	5.84	3.86	5.25
Avg. FR for Skin Color Calibration	1.45	1.51	1.57	1.46	1.25	1.55
Success						
Detection Rate	96.21%	92.89%	95.25%	93.88%	78.33%	85.66%
DR for CPE without WC	96.21%	98.14%	94.62%	55.94%	0%	80.63%
DR for CPE with WC	27.02%	46.53%	41.61%	7.59%	0%	10.94%
# of Failed Frames and Reasons						
Near Fingers Misleading	8	3	-	11	-	-
Weak Segmentation of Skin Color	23	-	-	28	13	24
Wrong Localization of Wrist	-	8	3	-	-	-
Ambiguity in Peak Determination	-	24	27	3	-	-
Extreme Lightning	-	7	-	-	-	14

According to the results in table 6.4 the success rate of the algorithm does not depend on the used camera because the first two video were recorded with different cameras and the results were similar. The success rate basically decrease if there with weak segmentation of the skin color or if the user is not very clear to the camera while he is showing his fingers to the camera. 'Near fingers misleading' is meaning the unclear determination of the fingers even with eyes. If the user is distant from the camera or if the user is holding his fingers very near to each

other this type of failure occurs. Weak segmentation of skin color is the biggest deficiency of the implemented system. It clearly occurs if the illumination condition is change during video. Change in lightning position or a significant change in the angle of the hand posture may result in an important change in the luminance feature of the skin color. Wrong localization of the wrist is usually occurs if the user is holding his hand in an angle. Then the middle point of the hand-wrist intersection line might be misaligned. Ambiguity in peak determination occurs if the users hand posture is leading some misleading peaks in the histogram or small noise fluctuations may result such a case. Extreme lightning is corresponding to the nonlinear acquisition behavior of the camera. Camera lens may yield misleading information for extreme bright or dark pixels. Here is the distribution of the failures of the system in the experiments. The distribution of the failure reasons is in table 6.5.

Table 6.5: Failure Reasons Distribution

Failure Reason	**Portion**
Near Fingers Misleading	11%
Weak Segmentation of Skin Color	45%
Wrong Localization of Wrist	5%
Ambiguity in Peak Determination	27%
Extreme Lightning	12%

The system works nearly in real time. The experiments were performed in a Core Duo Computer with 1.5 GB RAM. A higher configuration system might work the implementation in more than 15 fps which satisfies minimum real time condition. The overall success rate for all the frames in the experiments is 93.57%. Most of the goals in the beginning of this study are achieved. System works without using any special gloves or attached device, it has a tolerable failure rate, it works with different skin colored users and it does not depend on the camera parameters. However change in illumination condition is still a big handicap for the system to work properly for all lightning conditions. Also peak determination is another problem which should be solved with future studies. These issues will be discussed in chapter 7.

6.4 Application of The Theory: Remote Media Player

Media player is one of the significant applications where the comfort of the user increases excessively if it is controlled by hand gestures remotely. Users usually starts a media in a

computer sits in a distant position from the computer and watches. Increasing, decreasing volume, pausing, moving forward, changing the video etc. all needs the user input. If a user can do such things from distant without going near to the keyboard, it would be more comfortable. Remote controllers can perform such interface issues but one needs to pay much to introduce them to a PC or laptop because they are not normally integrated in commercial use PCs. Also instead of hand gestures, sitting with a remote control would be less comfortable in such a case.

For Remote Media Player application an interface is constructed in *MacroMedia Flash Player* to have an esthetic media player interface. The system in the background segments the skin color for the current condition and recognize the hand gestures. This background system was written in *MATLAB* and deployed to a *.NET* dll library file. Finally the integration between the Flash Player and the background gesture recognizer is done through an application written in *Visual Studio* 2005 using *C#* language. An instance of the application is shown in figure 6.5.

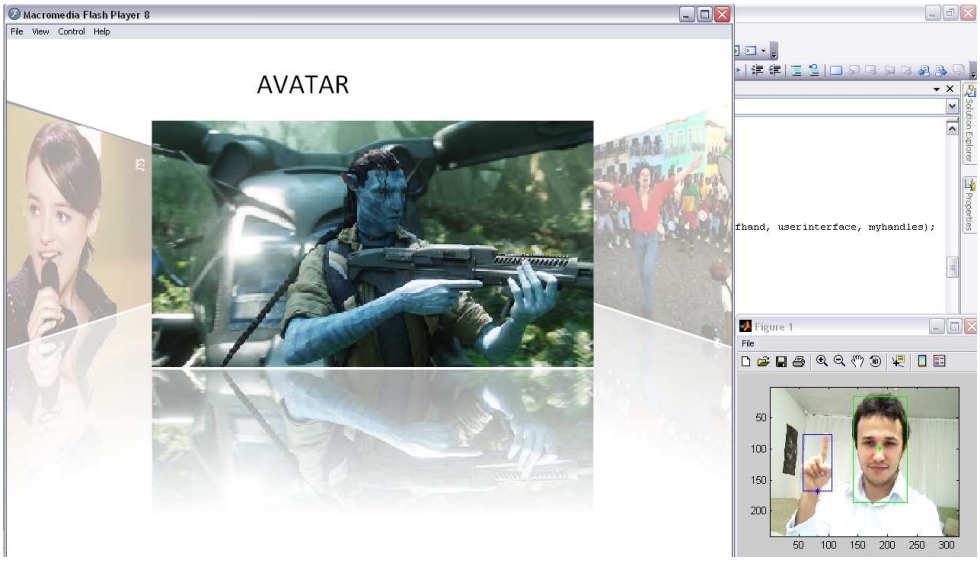
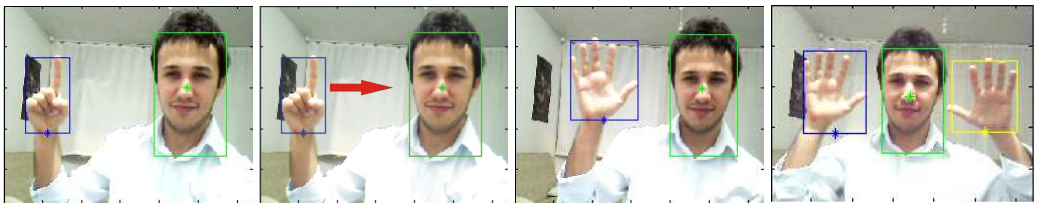


Figure 6.5: Instance of the Remote Media Player Application.

As shown in figure 6.5 user interacts with the media player by using the hand gestures. Each interaction needs a special gesture. Showing just one finger means to choose a video. While one is shown to the camera moving the hand in a way will make the media player to choose another movie. Showing five to the camera would start the selected video in full screen mode.

While the movie is running a left hand index finger means volume up and a right hand index finger means volume down. Similarly left hand palm will pause and right hand palm will resume the video. If both hands show palm at the same time then the video will stop and media list will appear again. Some of the defined gestures for the media player are illustrated in figure 6.6.



(a) Select video (b) Move to another video (c) Start selected video (d) Stop video

Figure 6.6: Some of the remote media player gestures.

Dynamic gestures are typically recognized by extracting the displacement vector of the hand center. If the displacement vector is bigger than a threshold value in a defined direction and if the gesture is a distinct one at the time of displacement then a predefined media player function is called. Typical displacement vector between two cascading frames is extracted by the following formula.

$$DV_n = HandCenter_n(x, y) - HandCenter_{n-1}(x, y) \tag{6.1}$$

Where DV_n is the displacement vector for the n_{th} frame. Displacement vectors are added for cascading last 5 frames and if the magnitude of resultant vector is bigger than .1 of the image width, then it is checked that if a dynamic gesture was performed or not.

6.5 Application of The Theory: 3D Flight War Game

Hand gesture recognition technology have started playing an important role in game industry. Major game console manufactures are introducing hand and body gestures based games in their new systems. Although there is not a perfect game system based on hand gestures working error-free, playing in a controller free environment is taking people's attention. As an application of this thesis study a flight war game was implemented. Game was built in *Visual*

Studio 2005 using *DirectX* and *.NET* libraries. It uses a background system developed in MATLAB to recognize hands, which was used in Remote Media Player also. For this war game, user use both hands as punch position. If hands stay in the same vertical line then the plane goes straight, if they are both in above face then plane goes upwards. if left hand is above face and right is below face then plane turn right etc. To fire and hit balloons user just needs to show palms to the camera. An instance of the game is shown in figure

Figure 6.7: Instance of the 3D Flight War Game.

CHAPTER 7

CONCLUSIONS

In this thesis study, a hand gesture recognition system which works under all lightning conditions with different skin colored users and with different camera parameters was aimed. It should not need any training or not make the user wear a special glove etc. Also the system was aimed to work in or nearly real time to be applicable in human computer applications. Finally it should work in a typical PC with a cheap USB webcam.

In the experiments, we could have a working system with the mentioned theory. However it has still some deficiencies and not working in 100% performance. First of all. it was observed that extreme lightning conditions always yielded non-linearities in acquired image pixel information. This dropped the performance of our algorithm especially when the extreme lightning conditions are not uniform on hand and face. If a sun light is directly coming into a room and lightning just one side of the user it would cause misleading skin color calibration. For this reason, we avoided to have non-uniform extreme lightning conditions in our experiments. If the lightning is not extremely shinny or dark, then the skin color could be calibrated correctly.

For the hand segmentation part we have used an adaptation of the method mentioned in [2]. According to the method $nRGB$ color space skin locus is used to eliminate the non-skin pixels in a coarse manner and than a special fine skin color extraction is used to extract the real skin pixels for the current conditions as mentioned in chapter in 4. In this process, the biggest problem was caused by the non-perfect elimination of the luminance components in pixels. Although HSV, $nRGB$ and $YCbCr$ is supposed to eliminate brightness, however it is known that brightness feature can be eliminated up to a point as mentioned in literature review of this thesis. Once the illumination is soft, it was very easy to segment hand correctly, but in extreme lightning conditions we have come up with some extra correction algorithms. For

instance, if there is a very bright light shining from just one side of the user, then that side of the arm may not fall in skin locus due to the non-linearity of the camera for extreme lightning conditions and due to non-perfect elimination of brightness in the color space. Then, wrist cutting procedure might yield unwanted results. Wrist cutting is started from the arm and goes to the hand with counting the skin pixels at each step. Since hand is thicker than the wrist if the skin pixels are increased significantly in a step, then the wrist is found and the hand is cut at this point. But if there are false negatives in the arm due to the mentioned reasons, than that intersection line might be in a misleading position. For such cases small tricks are used like, comparing the face size with the hand size to check if an abnormality occurs or not. Such an instance frame is shown in figure 7.1. As clear seen in figure there is significant portion of false positive skin pixels on the left hand side of both face and arm. This causes the false detection of the hand. Blue box encloses a bigger part than the hand and the centroidal profile extraction starts at a wrong point. Since the thumb is not in the direction of the starting point of the profile extraction, it counts 4 fingers missing the thumb.

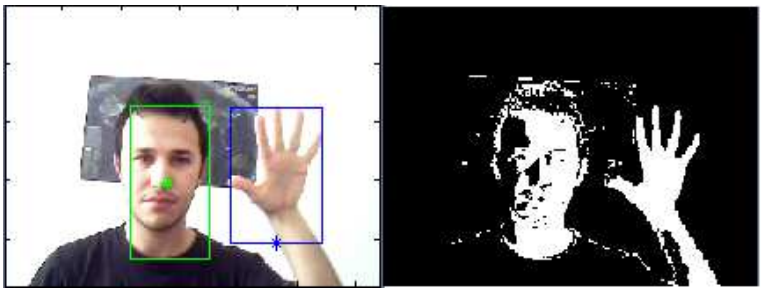


Figure 7.1: Extreme lightning case.

Another basic problem is about the peak detection on the histograms. Peaks sizes are proportional to the size of the skin pixels in both g_{neg}, g_{pos} histograms and centroidal profile extraction histograms. If palm is shown to the camera (showing 5), then thumb might be in a misleading position. If it is wide apart from the other 4 fingers centroidal profile extraction might not count the peak as a histogram or the reverse procedure might take place. Just 4 fingers are shown to the camera and the small local peak might be faulty in such a case if it is considered as thumb. Such uncertain histogram distribution is illustrated in figure 7.2. Both histograms are taken from a palm image. In spite of the peaks of histogram on the left can be extracted clearly, shown peak is hard to interpret if it is an actual thumb or just a noise.

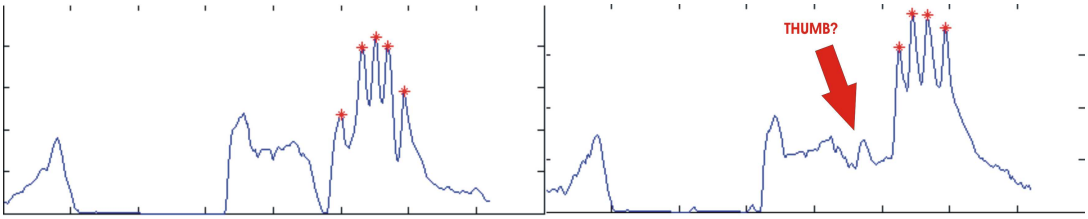


Figure 7.2: Uncertain histogram peaks.

Another failure reason is that having very skin like materials in the background. Some sort of wooden tables may be so similar to skin and it might not be possible to distinguish it from real skin pixels. Our system does not have any background segmentation procedure, so the system fails if such a background exists.

Some certain cases are left as don't care conditions, like collision of face and hands or very near finger separations. These cases are out of the scope of this thesis study. Also the transition frames are considered as don't care frames because while the user is rising a finger it is not possible to categorize the glance of that finger.

To measure the success rate of our system, a test bed was constructed as mentioned in chapter 6. Test videos are the same videos used to compare the skin color performances. Instances of those videos were shown in figure 6.1. Some of the videos are not challenging with simple backgrounds and with clear finger exhibitions. However some of the videos compel the success rate of the system. The success rate of the videos are summarized in table 6.4.

According to the test results in table 6.4 the success rate of the algorithm does not depend on the used camera or the skin color of the user. It basically decrease if there is an extreme lightning conditions or if the user is not very clear to the camera while he is showing his fingers to the camera. The system work nearly in real time. Most of the goals in the beginning of this study are achieved. System works without using any special gloves or attached device, it has a tolerable failure rate, it works with different skin colored users and it does not depend on the camera parameters. However extreme lightning condition is still a big handicap for the system to work properly for all lightning conditions.

As a future work, all the problems presented above might be solved. In summary, extreme lightning condition might be handled by deeper analysis or by trying different color spaces

rather than *nRGB* to see if there is an improvement or not. Also histogram peak ambiguity is another topic to have a perfect recognition rate. The system in this study is working nearly in real time but it can be fasten by implementing it in a more low level programming technique. Finally, applications can be improved in both performance and aesthetical manners to convert the system in a commercial product.

REFERENCES

[1] G. Wyszecki and W. S. Stiles, Color Science: Concepts and Methods, Quantitative Data and Formulas, John Wiley & Sons, Inc., London 1967

[2] Aryuanto Soetedjo, Koichi Yamada. Skin Color Segmentation Using Coarse-to-Fine Region on Normalized RGB Chromaticity Diagram for Face Detection., BIEICE Trans. Inf. & Syst., Vol.E91-D, No.10, October 2008.

[3] M. Storring, H. Andersen, and E. Granum. Skin colour detection under changing lighting conditions. Proc. 7th Symposium on Intelligent Robotics Systems, Coimbra, Portugal, July 1999.

[4] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, Skin detection in video under changing Illumination conditions, Proc. Computer Vision and Pattern Recognition, vol.1, pp.839-842, 2000.

[5] Jae-Ho Shin, Jong-Shill Lee, Se-Kee Kil, Dong-Fan Shen, Je-Goon Ryu, Eung-Hyuk Lee, Hong-Ki Min, Seung-Hong Hong. Hand Region Extraction and Gesture Recognition Using Entropy Analysis., IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.2A, February, 2006.

[6] Asanterabi Malima, Erol Özgür, and Müjdat Çetin. A Fast Algorithm For Vision-Based Hand Gesture Recognition For Robot Control.

[7] Moritz Störring, Thomas Moeslund, Yong Liu, and Erik Granum. Computer Vision-Based Gesture Recognition For an Augmented Reality Interface. In 4th IASTED International Conference on Visualization, Imaging and Image Processing, pages 766-771, Marbella, Spain, Sep 2004

[8] N. Soontranon, S. Aramvith, and T.H. Chalidabhongse. Face and Hands Localization and Tracking for Sign Language Recognition. International Symposium on Communications and Information Technologies 2004 (ISCIT 2004), Sapporo, Japan, October 26-29, 2004

[9] U. Ahlvers, U. Zölzer, R. Rajagopalan. Model-free Face Detection and Head Tracking with Morphological Hole Mapping, EUSIPCO'05, Antalya, Turkey.

[10] Oya Aran, Cem Keskin, Lale Akarun. Computer Applications for Disabled People and Sign Language Tutoring. Proceedings of the Fifth GAP Engineering Congress, 26-28 April 2006, Şanlıurfa, Turkey.

[11] B. Ionescu, D. Coquin, P. Lambert, V. Buzuloiu. Dynamic Hand Gesture Recognition Using the Skeleton of the Hand. EURASIP Journal on Applied Signal Processing 2005:13, 2101-2109, Hindawi Publishing Corporation.

[12] Oya Aran, Lale Akarun. Recognizing Two Handed Gestures with Generative, Discriminative and Ensemble Methods via Fisher Kernels. Multimedia Content Representation, Classification and Security International Workshop, MRCS 2006, İstanbul, Turkey.

[13] Aykut Tokatlı, Uğur Halıcı. 3D Hand Tracking in Video Sequences. MSc Thesis, September 2005, Middle East Technical University.

[14] M. H. Yang, D. J. Kriegman, N. Ahuja. Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, pp. 34-58, January 2002.

[15] M. H. Yang, N. Ahuja. Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition Proceedings of the CVPR, pp. 892-897, Santa Barbara, 1998.

[16] R. Hassanpour, A. Shahbahrami, S. Wong. Adaptive Gaussian Mixture Model for Skin Color Segmentation. Proceedings of World Academy of Science, Engineering and Technology Volume 31, July 2008.

[17] F. Porikli, T. Haga. Event Detection by Eigenvector Decomposition Using Object and Frame Features. Conference on Computer Vision and Pattern Recognition (CVPRW), Vol. 7, pp. 114, June 2004.

[18] W. H. Andrew Wang, C. L. Tung. Dynamic Hand Gesture Recognition Using Hierarchical Dynamic Bayesian Networks Through Low-Level Image Processing. Peoceedings of the Seventh International Conference on Machine Learning and Cybetnetics. Kunming, 12-15 July 2008.

[19] S. Marcel , O. Bernier , J. E. Viallet , D. Collobert, Hand Gesture Recognition Using Input-Output Hidden Markov Models. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000, p.456, March 26-30, 2000

[20] Lu Huchuan, Shi Wengang. Skin-Active Shape Model for Face Alignment. Proceedings of the Computer Graphics, Imaging and Vision: New Trends, CGIV'05, 2005.

[21] Yao-Jiunn Chen, Yen-Chun Lin. Simple Face-detection Algorithm Based on Minimum Facial Features. The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Taipei, Taiwan, Nov. 5-8, 2007.

[22] M. J. Jones, Daniel Snow. Pedestrian Detection Using Boosted Features over Many Frames. International Conference on Pattern Recognition (ICPR), Motion, Tracking, Video Analysis, December 2008.

[23] P. Chakraborty, P. Sarawgi, A. Mehrotra, G. Agarwal, R. Pradhan. Hand Gesture Recognition: A Comparative Study. Proceedings of the Internatinaol MultiConference of Engineers and Computer Scientists 2008 Vol 1, IMECS 2008, Hong Kong, 19-21 March 2008.

[24] L. Sabeti, Q. M. Jonathan Wu. High-Speed Skin Color Segmentation for Real-Time Human Tracking. IEEE Internatinonal Conference on Systems, Man and Cybernetics, ISIC2007, Montreal, Canada, 7-10 Oct. 2007.

[25] C. H. Kim, J. H. Yi. An Optimal Chrominance Plane in the RGB Color Space for Skin Color Segmentation. International Journal of Information Technology vol.12 no.7, pp.73-81, 2006.

[26] S. Askar, Y. Kondratyuk, K. Elazouzi, P. Kauff, O. Scheer. Vision-Based Skin-Colour Segmentation of Moving Hands for Real-Time Applications. Proc. Of. 1st European Conference on Visual Media Production, CVMP, London, United Kingdom, 2004.

[27] A. Albiol, L. Torres, E. J. Delp. An Unsupervised Color Image Segmentation Algorithm For Face Detection Applications. Proc. of International Conference on Image Processing 2001, 7-10 Oct 2001.

[28] A. Hadid, M. Pietikainen, B. Martinkauppi. Color-Based Face Detection Using Skin Locus Model and Hierarchical Filtering. Proc. of 16th International Conference on Pattern Recognition 2002, vol4, pp. 196-200, 2002.

[29] J. C. Terrillon, S. Akamatsu. Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images. Proc. of Vision Interface '99, pp. 180-187, Trois-Rivieres, Canada, 19-21 May 1999.

[30] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt. A Skin Tone Detection Algorithm for an Adaptive Approach to Steganography. Signal Processing, v.89 n.12, p.2465-2478, December, 2009.

[31] L.H. Zhao, X.L. Sun, J.H. Liu, X.H. Xu. Face Detection Based on Skin Color. Proc. International Conference OnMachine Learning and Cybernetics, vol.6, pp.3625-3628, Shanghai, China, August 2004.

[32] Y. Freund and R. Schapire. A decision-theoritic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci., vol.55 no.1, pp. 119-139, August 1997.

[33] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, 2001.

[34] Ahmet Bahtiyar Gül, Aydin Alatan. Holistic Face REcognition by Dimension Reduction. MSc Thesis, September 2003, Middle East Technical University.

[35] Zhe Lin, Larry S. Davis, Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32 no.4, pp. 604-618, April 2010.

[36] R. R. Anderson, J. Hu, J. A. Parrish. Optical Radiation Transfer in the Human Skin and Applications in Vivo Remittance Spectroscpoy. In R. Marks and P. A. Payne, editors, Bioengineering and the Skin, MTP Press Limited, chap. 28, pp.253-265. 1981.

[37] S. Jayaram, S. Schmugge, M. C. Shin, L. V. Tsap. Effect of Colorspace Transformation, The Illuminance Component, and Color Modelling on Skin Detection. Proc. of IEEE Computer Vision and Pattern Recognition (CVPR'04), Vol. 2, pp. 813-818, Washington, USA, 27th June-2nd July 2004.

[38] K. C. Yow, R. Cipolla. Feature-Based Human Face Detection. Image and Vision Computing, vol. 15, no. 9, pp.713-735, 1997.

[39] Bauer, Hienz. Relevant feature for video-based continuous sign language recognition. Department of Technical Computer Science, Aachen University of Technology, Aachen, Germany, 2000.

[40] Starner, Weaver, Pentland. Real-time American sign language recognition using a desk and wearable computer-based video. In proceedings IEEE transactions on Pattern Analysis and Machine Intelligence, pages 1371-1375, 1998.

[41] Bowden, Sarhadi. Building temporal models for gesture recognition. In proceedings British Machine Vision Conference, pages 32-41, 2000.

[42] Davis, Shah. Visual gesture recognition. In proceedings IEEE Visual Image Signal Process, vol.141, No.2, pages 101-106, 1994.

[43] R. Lockton, A. W. Fitzgibbon. Hand Gesture Recognition Using Computer Vision. BSc. Graduation Project, Oxford University.

[44] Chan Wah, S. Ranganath. Real-time gesture recognition system and application. Image and Vision Computing 20, pp. 993-1007, 2002.

[45] Attila Licsar, Tamas Sziranyi. User-adaptive hand gesture recognition system with interaction training. Image and Vision Computing 23, pp. 1102-1114, 2005.