



Remote Surveillance and Face Tracking with Mobile Phones (Smart Eyes)

by

Sandro da Silva



A thesis submitted in partial fulfillment of

the requirements for the degree of

Magister Scientiae

in the Department of Computer Science

University of the Western Cape

Lead Investigator: Prof. Johnson I. Agbinya

May 2005

Remote Surveillance and Face Tracking with Mobile Phones (Smart Eyes)

Sandro da Silva

KEYWORDS

Remote Surveillance, Face Tracking, Mobile Phones, Face Detection, HSV Color Space,
Face Recognition, Color Histogram, Skin Color Pixel, intensity gradient.

ABSTRACT

Remote Surveillance and Face Tracking with Mobile Phones

By: Sandro da Silva
M.Sc. Thesis
Department of Computer Science
Faculty of Science
University of the Western Cape

This thesis addresses analysis, evaluation and simulation of low complexity face detection algorithms and tracking that could be used on mobile phones. Network access control using face recognition increases the user-friendliness in human-computer interaction. In order to realize a real time system implemented on handheld devices with low computing power, low complexity algorithms for face detection and face tracking are implemented. Skin color detection algorithms and face matching have low implementation complexity suitable for authentication of cellular network services. Novel approaches for reducing the complexities of these algorithms and fast implementation are introduced in this thesis. This includes a fast algorithm for face detection in video sequences, using a skin color model in the HSV (Hue-Saturation-Value) color space. It is combined with a Gaussian model of the H and S statistics and adaptive thresholds. These algorithms permit segmentation and detection of multiple faces in thumbnail images. Furthermore we evaluate and compare our results with those of a method implemented in the Chromatic Color space (YCbCr). We also test our test data on face detection method using Convolutional Neural Network architecture to study the suitability of using other approaches besides skin color as the basic feature for face detection. Finally, face tracking is done in 2D color video streams using HSV as the histogram color space. The program is used to compute 3D trajectories for a remote surveillance system.

DECLARATION

I declare that *Remote Surveillance and Face Tracking with Mobile Phones (Smart Eyes)* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.



Sandro Cahanda Marinho da Silva

May 2005

TABLE OF CONTENTS

KEYWORDS	2
Abstract	3
DECLARATION	4
TABLE OF CONTENTS	5
LIST OF FIGURES AND TABLES	7
ACKNOWLEDGEMENTS	8
CHAPTER 1	9
INTRODUCTION TO THE RESEARCH	9
1.1 Introduction	9
1.2 Motivation	11
1.3 Accomplishments	11
1.4 Thesis Focus and Constraints	12
1.5 Conclusion	13
CHAPTER 2	14
LITERATURE REVIEW	14
2.1 Introduction	14
2.2 Overview of Remote Surveillance Systems	15
2.3 Face Detection: Background and Related Work	15
2.3.1 Knowledge-based methods	18
2.3.2 Feature Invariant Approach	19
2.3.2.1 Skin Color Modelling	19
2.3.3 Template Matching	21
2.3.4 Appearance-Based Methods	22
2.4 Face Recognition Systems: Background and Related Work	24
2.4.1 Face recognition from image sequences	24
2.4.1.1. Basic Techniques of Video-Based Face Recognition	25
2.4.2.2 Video-Based Face Recognition	30
2.5 Summary	38
CHAPTER 3	39
FACE DETECTION USING HSV SKIN COLOR MODELING	39
3.1 Introduction	39
3.2 HSV Skin color model	41
3.3 Skin Color Segmentation	44
3.4 Selection of face regions	45
3.5 Generation of Average Face Model	48
3.6 Experimental Results	50
3.6.1 Convolutional Neural Network	53
3.6.2 Computing Time	56
3.7 Discussion	57
3.8 Summary	59
CHAPTER 4	59
FACE DETECTION USING YCbCr SKIN COLOR MODELING	59
4.1 Introduction	60
4.2 YCbCr Skin Color Model	61
4.3 Skin Color Segmentation	64

4.4 Selection of Face Regions	66
4.5 Generation of Average Face Model	67
4.6 Experimental Results	67
4.6.1 Computing Time.....	71
4.7 Discussion	71
4.8 Comparison between the HSV approach and YCbCr technique	72
4.9 Summary	77
CHAPTER 5	78
FACE TRACKING	78
5.1 Introduction	78
5.2 Face Tracking Method	79
5.3 Experimental Results	82
5.4 Discussion	84
5.5 Comparison with other Face Tracking Approaches	84
5.6 Conclusion	85
CHAPTER 6	86
CONCLUSION AND DIRECTIONS FOR FURTHER RESEARCH	86
6.1 Introduction	86
6.2 Research Questions	87
6.3 Observations	92
6.4 Applications	94
6.5 Directions for Further Research	94
References.....	96



LIST OF FIGURES AND TABLES

Figure 1.1: Real case scenario for a remote surveillance system with mobile phones.....	10
Table 2.1: Categorization of Video-Based Face Recognition Techniques.....	30
Figure 3.1: The HSV Color Model.....	41
Figure 3.2: Shows the Skin sample in the RGB space and the HSV space.....	42
Figure 3.3: Color distribution in the HSV color space for skin color of different people of various ethnic groups.....	43
Figure 3.4: Fitting skin color into a Gaussian distribution model.....	44
Figure 3.5: shows the RGB image and the equivalent HSV image.....	45
Figure 3.6: shows the original image next to the skin-likelihood image.....	45
Figure 3.7: Shows the skin-likelihood image and the skin-segmented image.....	46
Figure 3.8: Generation of Average Face Template (model).....	50
Figure 3.9: General Overview of the described system.....	51
Figure 3.10: (a) The original color images; (b) Skin-likelihood images; (c) skin-segmented images; (d) final results.....	52
Table 3.1: Performance of algorithm with HSV model.....	53
Figure 3.11: Test results of a NN approach and HSV model.....	56
Figure 4.1: Shows the Skin sample in the RGB space and the YCbCr space.....	63
Figure 4.2: Color distribution in the chromatic color space for skin color of different people.....	64
Figure 4.3: Fitting skin color into a Gaussian distribution model.....	65
Figure 4.4: shows the RGB image and the equivalent YCbCr image.....	66
Figure 4.5: shows the original image next to the skin-likelihood image.....	66
Figure 4.6: Shows the skin-likelihood image and the skin-segmented image.....	67
Figure 4.7: General Overview of the described System.....	68
Figure 4.8: (a) The original color images; (b) Skin-likelihood images; (c) skin-segmented images; (d) final results.....	70
Table 4.1: Performance of algorithm with YCbCr model.....	71
Figure 4.9: Skin data (blue) vs. background data (red) in HS and HV color space.....	74
Figure 4.10: Skin data (blue) vs. background data (red) in CbCr color space.....	74
Figure 4.11: Skin data (blue) vs. background data (red) in S-V color space.....	75
Figure 4.12: Plot of the skin pixel samples and the bounding equations.....	75
Figure 4.13: Experimental results of how the HSV method handles skin-color-like pixels as compared to the YCbCr method.....	76
Figure 4.14: Skin (blue) vs. hair (red) pixels.....	77
Figure 5.1: Some frames of the image sequence shown below.....	83
Figure 5.2: Some frames of the image sequence shown below.....	84

ACKNOWLEDGEMENTS

My thanks and appreciation goes to our dear Father in Heaven, the Lord God Almighty, for having my best interest at heart and for being the God of comfort, health, strength and opportunities.

I also would like to convey my most profound appreciation to my supervisor Prof. Johnson I. Agbinya for his valuable contribution and expertise. His mentorship contributed to the completion of this dissertation and to my personal development.

Further we would like to thank Javier Martinez from the University of Nevada, currently a Research Assistant in the Computer Vision Lab in the Department of Computer Science in Reno, for his assistance with the tracking.



A special word of appreciation goes to my parents Joao and Emilia for always believing in me and for giving me the wings to fly. My brother Ivan, my sisters Maricel and Camy for all their love.

I am also grateful to Mmalewane, Palesa, Wilson, Dele, and Mlungisi for being great friends in God's campus ministry, and I am also indebted to Thabiso for being a best friend. Lucia Pires at Career Wise for all your help.

My fellow students in the yellow submarine for their camaraderie.

Finally a sincere debt of gratitude to the Angolan Ministry of Petroleum, and to the Telkom Centre of Excellence (CoE) for the funding that enabled me to pursue this research endeavor.

CHAPTER 1

INTRODUCTION TO THE RESEARCH

1.1 INTRODUCTION

Telecommunication services are evolving towards all IP services on IP enabled networks.

Existing 2G mobile networks are FDMA, TDMA (GSM) or CDMA based offering data rates well below PC modems. A generation of fast, data-rich, IP and multimedia services accessed instantly over mobile handsets is emerging. The capacity of new networks such as 3G and 4G technologies will allow operators to use data rates of up to 2 Mbps or even more on wireless networks. In the evolution of 2G to 3G networks, some operators will upgrade and adapt their existing 2G networks whereas others will build completely new ones.



A major driver behind building 3G networks is to increase revenue through new data services, to counteract the gradual decrease of voice revenue. To achieve this, these data services (IP and multimedia services) have to be increased through the offering of rich content to mobile handsets and wireless devices. Increasing capabilities of mobile handsets demands the provision of new applications to offer such services.

Therefore, we consider that home and property wireless surveillance systems, particularly those that can be monitored remotely would be some of such applications.

The face provides a variety of different communicative functions such as identification, the perception of emotional expressions. Biometric systems using face recognition are attracting attention for authentication and authorization. Similarly, the cost effectiveness of sending only a facial image over a narrowband wireless network instead of a full size photo can hardly be contented. Face detection and tracking of human

faces is the main objective of this research project, which would benefit the surveillance process.

However, in order to realize a real time system on handheld devices with low computing power and memory capacity, low complexity algorithms for face capture, detection and tracking are required. This thesis addresses face detection on still images based on color techniques, it also addresses face tracking on video streams based on a method that combines the output of two modules; one that matches the intensity gradients along the face's boundary and one that matches the color histogram of the face's interior in the HSV color space.

Real case scenario:

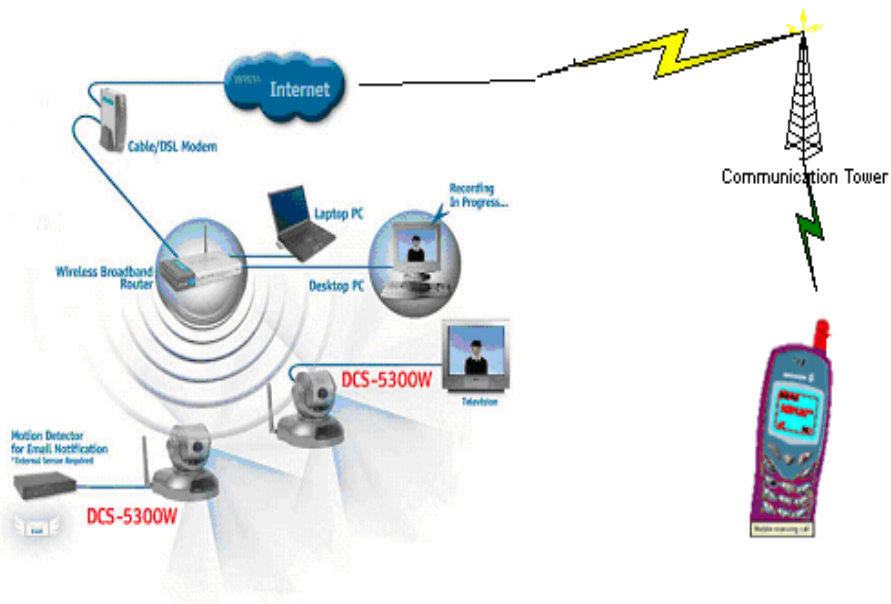


Figure 1.1: Real case scenario for a remote surveillance system with mobile phones

The real case scenario is illustrated in **figure 1.1**, and the aim is to deploy wireless video cameras, like DCS – 5300W, on a car or at home. When needed, the video cameras are triggered and immediately they start recording the video sequence of the intruder.

While recording, the system must then be able to detect and track the face of the intruder and act upon the results. The video sequence of the intruder is streamed to either one of the computing units for processing, such as mobile handset, desktop PC, laptop PC or television on a wireless network.

1.2 MOTIVATION

A new generation of fast, data-rich, multimedia services accessed instantly over mobile handsets is emerging worldwide. 3G is the technology that makes this possible. Every telecom operator, developer and vendor is going to be affected by this technology as telecommunications evolves towards a third generation of networks, services and applications.

The deployment of such advanced technology is not financially feasible if it is only to support voice traffic. Therefore there is a very high demand from network operators and vendors for IP and multimedia services that will optimize the use of this technology.

Because the system will be deployed in many cars, the data traffic generated be high and has therefore potential to generate revenue for an operator.

1.3 ACCOMPLISHMENTS

We applied Skin color modeling using the HSV color space for face detection on mobile devices for use in remote surveillance. Fast face detection can be used to initialize face tracking. With the results obtained from the face detection phase we implemented face tracking using two modules that work orthogonally so that when one fails the other succeeds. The first module uses color histogram in HSV color space whereas the second one uses the image gradient.

It is possible to implement a mobile phone-based security system that will monitor intrusion and access to cars, even though there are some key issues to solve, like whether

the power budget of mobile handsets can handle processor-intensive video work, and whether the low picture quality of video telephony can be improved. Moreover, we give the reader some insight into skin color modeling approach for face detection using either the HSV color space or YCbCr color space and supply sufficient motivation for using skin color as an adequate solution to our problem statement. The performance levels attained were sufficiently satisfactory to prove the feasibility of applying skin color modeling with the HSV color for face detection as a prior step for face tracking on a mobile phone-based security system that will monitor the intrusion and access to cars.

1.4 THESIS FOCUS AND CONSTRAINTS

The objective of this thesis is to research and implement a mobile phone-based car security system that could monitor intrusion and access to cars. To implement a real-time face tracking system on handsets, we need to search for low complexity algorithms for face detection and face tracking, implement and test them. We need to examine the existing tracking and detection methods, analyze their strengths and weaknesses and rate of accuracy and find out a way of implementing them on handheld devices.

This thesis proposes the use of skin color modeling as the basic feature for the face detection approach and for face tracking we propose two orthogonal modules, one using color histograms and the other using the intensity gradient of the images.

These approaches have been successfully used in many applications of computer vision, but the same algorithms were not directly suitable for mobile devices.

The specific research questions that guided this study are as follows:

- 1) What are the advantages and disadvantages of having the face tracking system either on the hand held or on a remote server to which the device has access to?
- 2) Does the color of the face affect the tracking results?
- 3) Does the time of the tracking influence the tracking results?

- 4) Can we network the surveillance cameras?
- 5) What triggers the surveillance cameras?
- 6) How should we initiate automatic triggering to avoid false alarm?

1.5 CONCLUSION


In this chapter we have given an introduction and motivation behind this research endeavor; we have described our accomplishments and the focus and constraints for this thesis.

In chapter two of this dissertation we describe the literature review, where we give an overview of remote surveillance systems, face detection systems and finally we present some face recognition approaches on image sequences.

In chapter three we describe our proposed solution for face detection using HSV skin colour modeling. This technique uses a skin model to determine the most likely skin region, those skin regions are evaluated individually to determine whether it is face or not. It uses a template face to evaluate whether the segmented skin region is a face. This skin model is made of skin pixels of people of various races. If a positive match is encountered a box is drawn around the face on the test image. The experimental results for this method are also presented in this chapter. This program is capable of detecting multiple instances of faces in a color image with varying illumination conditions and inhomogeneous background. This method is used as a prior step for the face tracking system described in chapter six. It helps us to compute the initial coordinates of the face and the size of the face.

Besides using color as the basic feature for face detection, we thought it would be worthwhile to investigate another face detection technique as a possible solution to our problem. Thus, at the end of chapter three we discuss and present the experimental results for a face detection method using Convolutional Neural Network architecture.

Chapter four presents an alternative solution, where we use an algorithm that creates a skin color model in the YCbCr color space of people of different ethnic groups, and then we use a low pass filter to remove the effects of noise. This skin model is also used to determine the likelihood of the pixel values and segment the image into skin regions and non-skin regions, then we evaluate the skin regions individually for instances of faces. We also perform a very concise comparison between the method described in chapter three and this method.

In chapter five we describe our proposed solution for the face tracking problem for use in remote surveillance with mobile phones. The face's projection onto the image plane is modeled as an ellipse whose position and size are initially computed using the detection method presented on chapter three, then they are continually updated by a local search combining the output of a module focusing on the intensity gradient around the ellipse's perimeter with that of a module  on the color histogram, in the HSV color space, of the ellipse's interior. The experimental results presented indicate that it is a robust, real-time system that is able to track the face with enough accuracy.

Finally in chapter six we have discussion and conclusion and give some recommendations and directions for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The following session in this chapter discusses the state of the art of remote surveillance systems. The next session describes the background of face detection systems and a survey of different face detection approaches is presented in detail.

Considering that fast face detection is used as a prior step for face tracking, we therefore discuss the state of the art of face tracking techniques in the following session. Finally, we present a detailed evolution of mobile phones.

2.2 OVERVIEW OF REMOTE SURVEILLANCE SYSTEMS

Remote surveillance has proven to be a popular alternative to in-house security. Even if one pays employees to monitor security systems, it can often be a big help to dial-in and see your property. Remote surveillance is simple to manifest – it only involves linking your security camera to a PC.

An innovative work in this area is @Home – Remote Monitoring Patients [1]. It allows for health monitoring of patients by doctors in real time. It features a fully automated monitoring/surveillance system with a user-friendly interface. The system automatically triggers the alarm if the readings of the patient show an abnormality.

The health monitoring sensors measure health parameters such as blood pressure, pulse rate, temperature, respiration frequency, patient's weight, patient's motion, etc. there are also dispensers that provide the patient with the prescribed medication and record whether they were taken or not. The recordings from each medical sensor are transmitted to the clinic in real time via a wireless link and UMTS. The system is linked to the patient's PC in order to establish a videoconference between the patient and the medical team.

2.3 FACE DETECTION: BACKGROUND AND RELATED WORK

Definition of face detection: Given an arbitrary image, the aim of face detection is to find a face in the image and, if present, return the location of the image and extent of each face [2].

Face detection has a number of applications, namely it can be part of face recognition, a surveillance system, or video based computer machine interface.

Efficient face detection at frame rate is an impressive goal; it is analogue to face tracking that requires no knowledge of previous frames. Fast face detection has an apparent application to practical face tracking in the sense that it can be used to initialize tracking.

Face detection is an essential research problem because it has a role as a challenging case of a more general problem, i.e. object detection. Face detection is a beautiful paradigm to the general problem, because a face is naturally recognizable to a human being despite its many points or variations (e.g. skin tone, hairstyle, glasses, etc). Human beings are able to detect a face in the context of an entire person, but we want a simple, context-free approach to detection. Another source of difficulty for faces is the complex 3-dimensional shape of faces, and the resulting difference in the appearance of a given face under different lighting condition [3]. Therefore, a method that works well for faces can generally be trusted with the task of detection for a wide and complex object structures. The challenge associated with face detection can be attributed to the following factors:

- Pose.
- Presence or absence of structural components.
- Facial expression.
- Occlusion.
- Image orientation.
- Imaging conditions.

There are many closely related problems of detection. The goal of face localization is to determine the image position of a single face; this is a simplified detection problem

with the assumption that an input image contains only one face [4], [5]. The aim of *facial feature detection* is to detect the presence and location of features such as eyes, nose, nostrils, eyebrow, mouth, lips, etc., with the assumption that there is only one face in an image [6], [7]. *Face recognition* or *face identification* compares an input image (probe) against a database (gallery) and reports a match, if any [8], [9], [10]. The purpose of *face authentication* is to verify the claim of the identity of an individual in an input image [11], [12], while *face tracking* methods continually estimate the location and possibly the orientation of a face in an image sequence in real time [13], [14], [15]. *Facial expression recognition* concerns with identifying the affective states (happy, sad, etc.) of humans [16], [17]. However, *face detection* is the first step in any automated system, which solves the above problems.

The existing techniques to detect faces from a single intensity or color image are divided into four major categories:



1. Knowledge-based methods
2. Feature invariant approaches
 - Facial features
 - Texture
 - Skin color
 - Multiple features
3. Template matching methods
 - Predefined face templates
 - Deformable templates
4. Appearance-based methods
 - Eigenface

- Distribution-based
- Neural network
- Support Vector Machine (SVM)
- Hidden Markov Model (HMM)
- Information-Theoretical Approach

2.3.1 Knowledge-based methods

In this method, face detection methods are developed based on the rules derived from the researcher's knowledge of human faces. It is easy to come up with simple rules to describe the features of a face and their relationships. For example, a face often appears in an image with two eyes that are symmetric to each other, a nose, and a mouth. Their relative distances and positions can represent the relationships between features. Facial features in an input image are extracted first, and face candidates are identified based on the code rules. A verification process is usually applied to reduce false detections.

A problem with this technique is the difficulty in translating human knowledge into well-defined rules. If the rules are detailed, they may fail to detect faces that do not pass all the rules. The opposite case, they may give many false positives. It is also difficult to extend this approach to detect faces in different poses since it is challenging to enumerate all possible cases. On the other hand, heuristic about faces work well in detecting frontal faces in uncluttered scenes.

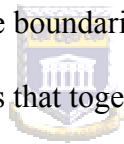
Yang and Huang used a hierarchical knowledge-based method to detect faces [18]. Their system consists of three levels of rules. At the highest level, scanning a window over the input image and applying a set of rules at each location to find all possible face

candidates. The rules at higher level are general description of what a face looks like while the rules at lower level rely on details of facial features.

Kotropoulos and Pitas [19] presented a rule-based localization method which is similar to [20] and [18].

2.3.2 Feature Invariant Approach

The underlying assumption is based on the observation that humans can effortlessly detect faces and objects in different poses and lighting conditions and, so, there must have properties or features that are invariant over these variabilities. Many methods have been proposed to first detect facial features and then to infer the presence of a face. Facial features are commonly detected using edge detectors. Based on the extracted features, a statistical model is built to describe their relationships and verify the existence of a face. One problem with these algorithms is that the image features can be corrupted due to illumination, noise, and occlusion. Feature boundaries can be weakened for faces, while shadows can cause numerous strong edges that together render perceptual grouping algorithms useless.



2.3.2.1 Skin Color Modelling

One of the invariant features of faces for detection that researchers have been trying to find is skin color.

Human skin color has been proven to be an effective feature in many applications from face detection to hand tracking. Although different people have different skin color, some studies have shown that the major difference lies largely between their intensity rather than chrominance [21], [22], [23]. Many color spaces have been used to label pixels as skin including RGB [24], [25], [26], normalized RGB [27], HSV (or HSI) [28], [29], [30], YCbCr [31], [32], YIQ [33], [34], YES [35] CIE XYZ [36] and CIE LUV [37].

Many methods have been proposed to build a skin color model. The simplest model is to define a region of skin tone pixels using Cr, Cb values [32], i.e., $R(Cr, Cb)$, from samples of skin color pixels. With chosen thresholds, $[Cr_1, Cr_2]$ and $[Cb_1, Cb_2]$, a pixel is classified to have skin tone if its values (Cr, Cb) fall within the ranges, i.e., $Cr_1 \leq Cr \leq Cr_2$ and $Cb_1 \leq Cb \leq Cb_2$.

Crowley and Coutaz used a histogram $h(r, g)$ of (r, g) values in normalized RGB color space to obtain the probability of obtaining a particular RGB-vector given that the pixel observes skin [38], [39]. In other words, a pixel is classified to belong to skin color if $h(r, g) \geq \tau$, where τ is threshold selected empirically from the histogram of samples. Saxe and Foulds proposed an iterative skin identification method that uses histogram intersection in HSV color space [28]. An initial patch of skin color pixels, called the control seed, is chosen by the user and is used to initiate the iterative algorithm. To detect skin color regions, their method moves through the image, one patch at a time, and presents the control histogram and the current histogram from the image for comparison. Histogram intersection [40] is used to compare the control histogram and current histogram. If the match score or number of instances in common (i.e., intersection) is greater than a threshold, the current patch is classified as being skin color.

Kjeldsen and Kender defined a color predicate in HSV color space to separate skin regions from background [29]. In contrast to the nonparametric methods mentioned above, Gaussian density functions [41], [42], [37] and a mixture of Gaussians [24], [25], [43] are often used to model skin color. The parameters in a unimodal Gaussian distribution are often estimated using maximum-likelihood [37], [43], [44]. The motivation for using a mixture of Gaussians is based on the observation that the color histogram for the skin of people with different ethnic background does not form a

unimodal distribution, but rather a multimodal distribution. The parameters in a mixture of Gaussians are usually estimated using an EM algorithm [24], [43].

Color information is an efficient tool for identifying facial areas and specific facial features if the skin color model can be properly adapted for different lighting environments. However, such skin color models are not effective where the spectrum of the light source varies significantly. This means that color appearance is often unstable due to changes in both background and foreground lighting. McKenna et al. presented an adaptive color mixture model to track faces under varying illumination conditions [44]. Instead of relying on a skin color model based on color constancy, they used a stochastic model to estimate an object's color distribution online and adapt to accommodate changes in the viewing and lighting conditions.

Skin color alone is not sufficient to detect or track faces. Several modular systems using a combination of shape analysis, color segmentation, and motion information for locating and tracking heads and faces in an image sequence have been developed [22], [37], [23], [44], [30].

2.3.3 Template Matching

In template matching, a standard face pattern is manually predefined by a function. Given an input image, the correlation values with the standard patterns are computed for the face contour, eyes, nose, and mouth independently. The correlation values determine the existence of a face. This approach is simple to implement, but it is inadequate for face detection since it cannot effectively deal with variation in scale, pose, and shape. Multiresolution, multiscale, subtemplates, and deformable templates have been proposed to achieve scale and shape invariance.


Sakai et al. [45] used several subtemplates for the eyes, nose, mouth, and face contour to model a face. Each subtemplate is defined in terms of line segments. Lines in the input image are extracted based on greatest gradient change and then matched against the subtemplates. The correlations between subimages and contour templates are computed first to detect candidate locations of faces. Then, matching with the other subtemplates is performed at the candidate positions. This means that the first phase determines focus of attention or region of interest and the second phase examines the details to determine the existence of a face. Later works on face detection has adopted this idea of focus of attention and subtemplates.

Yuille et al. [46] used deformable templates to model facial features that fit an a priori elastic model to facial features (e.g., eyes). Here, parameterized templates describe facial features. An energy function is defined to link edges, peaks, and valleys in the input image to corresponding parameters in the template. The best fit of the elastic model is found by minimizing an energy function of the parameters. Although their experimental results demonstrate good performance in tracking nonrigid features, one drawback of this approach is that the deformable template must be initialized in the proximity of the object of interest.

2.3.4 Appearance-Based Methods

The “templates” in appearance-based methods are learned from examples in images. Appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and nonface images. The learning characteristics are in the form of distribution models or discriminant functions that are consequently used for face detection. Meanwhile, dimensionality reduction is usually carried out for the sake of computation efficiency and detection efficacy.

Many appearance-based methods can be understood in a probabilistic framework. An image or feature vector derived from an image is viewed as a random variable x , and this random variable is characterized for faces and nonfaces by the class-conditional density functions $p(x|\text{face})$ and $p(x|\text{nonface})$. Bayesian classification or maximum likelihood can be used to classify a candidate image location as face or nonface. However, a straightforward implementation of the Bayesian classification is infeasible because of the high dimensionality of x , because $p(x|\text{face})$ and $p(x|\text{nonface})$ are multimodal, and because it is not yet understood if there are natural parameterized forms for $p(x|\text{face})$ and $p(x|\text{nonface})$. Hence, much of the work in an appearance-based method concerns empirically validated parametric and nonparametric approximations to $p(x|\text{face})$ and $p(x|\text{nonface})$.

An approach in appearance-based methods is to find a discriminant function (i.e., decision surface, separating hyper plane, threshold function) between face and nonface classes. Image patterns are projected to a  lower dimensional space and then a discriminant function is formed for classification [10], or a nonlinear decision surface can be formed using multiplayer neural networks [47]. Support vector machines and other kernel methods have also been proposed. These methods project patterns to a higher dimensional space and then form a decision surface between the projected face and nonface patterns [48].

In conclusion, face detection methods have been classified into four major categories. However, some methods can be classified into more than one category.

The common face detection methods are:

- a) Finding faces in images with controlled background;
- b) Finding faces by color;
- c) Finding faces by motion;

- d) Using a mixture of the above;
- e) Finding faces in unconstrained scenes.

2.4 FACE RECOGNITION SYSTEMS: BACKGROUND AND RELATED WORK

In any face recognition application, a face detection stage is needed, because fast face detection has an apparent application to practical face tracking in the sense that it can be used to initialize tracking and face tracking is a prior stage to face recognition.

Face recognition can be divided into two basic applications: identification and verification. In the identification problem, the face to be recognized is unknown and is matched against faces of a database containing known faces. In the verification problem the system confirms or rejects the claimed identity of the input face.

There are two major techniques for face recognition, namely face recognition from still images and face recognition from image sequences. On the next subsections we describe in detail face recognition from image sequences.



2.4.1 Face recognition from image sequences

A typical video-based face recognition system automatically detects face regions, extracts features from the video, and recognizes facial identity if a face is present. In surveillance, information security, and access control applications, face recognition and identification from a video sequence is an important problem. Face recognition based on video is preferable over using still images, since as demonstrated in [49] and [50], motion helps in recognition of (familiar) faces when the images are negated, inverted or thresholded. It was also demonstrated that humans could recognize animated faces better than randomly rearranged images from the same set. Significant challenges for video-based recognition still exist; we name several of them here.

1. The quality of video is low. Often, video acquisition occurs outdoors (or indoors but with bad conditions for video capture) and the subjects are not cooperative; hence there

may be large illumination and pose variations in the face images. In addition, partial occlusion and disguise are possible.

2.Face images are small. Again, due to the acquisition conditions, the face image sizes are smaller than the assumed sizes in most still-image-based face recognition systems. For example, the valid face region can be as small as 15 x 15 pixels¹, whereas the face image sizes used in feature-based still image-based systems can be as large as 128 x 128. Small-size images not only make the recognition task more difficult, but also affect the accuracy of face segmentation, as well as the accurate detection of the fiducial points/landmarks that are often needed in recognition methods.

3.The characteristics of faces/human body parts. One of the main reasons for the feasibility of generic descriptions of human behavior is that the intraclass variations of human bodies, and in particular faces, is much smaller than the difference between the objects inside and outside the class. For the same reason, recognition of individuals within the class is difficult. For example, detecting and localizing faces is typically much easier than recognizing a specific face.

There are three closely related techniques that are critical for the realization of the full potential of video-based face recognition, namely, face segmentation and pose estimation, face tracking, and face modeling. We are briefly going to review them before we examine existing video-based face recognition algorithms.

2.4.1.1. Basic Techniques of Video-Based Face Recognition

In [8], four computer vision areas were mentioned as being important for video-based face recognition: segmentation of moving objects (humans) from a video sequence; structure estimation; 3D models for faces; and nonrigid motion analysis. For example, in [25] a face modeling system was described. This system utilizes all four techniques:

¹ Notice this is totally different from the situation where we have images with large face regions but the final face regions feed into a classifier is 15 x 15.

segmentation of the face based on skin color to initiate tracking; use of a 3D face model based on laser-scanned range data to normalize the image (by facial feature alignment and texture mapping to generate a frontal view) and construction of an eigensubspace for 3D heads; use of structure from motion (SfM) at each feature point to provide depth information; and non-rigid motion analysis of the facial features based on simple 2D SSD (sum of square differences) tracking constrained by a global 3D model. Based on the current development of video-based face recognition, we think it is better to review three specific face-related techniques instead of the above four general areas. The three video-based face-related techniques are: face segmentation and pose estimation, face tracking, and face modeling.

(1) Face Segmentation and Pose Estimation

Early attempts [10] at segmenting moving faces from an image sequence used simple pixel-based change detection procedures based on difference images. These techniques may run into difficulties when multiple moving objects and occlusion are present. More sophisticated methods use estimated flow fields for segmenting humans in motion [51]. More recent methods [52], [53] have used motion and/or color information to speed up the process of searching for possible face regions. After candidate face regions are located, still-image-based face detection techniques can be applied to locate the faces [2]. Given a face region, important facial features can be located. The locations of feature points can be used for pose estimation, which is important for synthesizing a virtual frontal view [52]. Newly developed segmentation methods locate the face and estimate its pose simultaneously without extracting features [54], [55]. This is achieved by learning multiview face examples, which are labeled with manually determined pose angles.

(2) Face and Feature Tracking

After faces are located, the faces and their features can be tracked. Face tracking and feature tracking are critical for reconstructing a face model (depth) through SfM, and feature tracking is essential for facial expression recognition and gaze recognition.

Tracking also plays a key role in spatiotemporal-based recognition methods [56], [57] which directly use the tracking information.

In its most general form, tracking is essentially motion estimation. However, general motion estimation has fundamental limitations such as aperture problem. For images like faces, some regions are too smooth to estimate flow accurately, and sometimes the change in local appearances is too large to give reliable flow. Fortunately, these problems are alleviated thanks to face modeling, which exploits domain knowledge. In general, tracking and modeling are dual processes: tracking is constrained by a generic 3D model or a learned statistical model under deformation, and individual models are refined through tracking. Face tracking can be roughly divided into three categories: (a) head tracking, which involves tracking the motion of a rigid object that is performing rotations and translations; (b) facial feature tracking, which involves tracking nonrigid deformations that are limited by the anatomy of the head, that is, articulated motion due to speech or facial expressions and deformable motion due to muscle contractions and relaxations; and (c) complete tracking, which involves tracking both the head and the facial features.

Early efforts focused on the first two problems: head tracking [58] and facial feature tracking [59], [60]. In [58], an approach to head tracking using points with high Hessian values was proposed. Several such points on the head are tracked and the 3D motion parameters of the head are recovered by solving an over-constrained set of motion equations. Facial feature tracking methods may make use of the feature boundary or the

feature region. Feature boundary tracking attempts to track and accurately delineate the shape of the facial feature, for example, to track contours of the lips and mouth [80]. Feature region tracking addresses the simpler problem of tracking a region such as a bounding box that surrounds the facial feature [61].

In [61], a tracking system based on local parameterized models is used to recognize facial expressions. The models include a planar model for the head, local affine models for the eyes, and local affine models and curvature for the mouth and eyebrows. A face tracking system was used in [62] to estimate the pose of the face. This system used a graph representation with about 20-40 nodes/landmarks to model the face. Knowledge about faces is used to find the landmarks in the first frame. Two tracking systems described in [25] and [63] model faces completely with texture and geometry. Both systems use generic 3D models and SfM to recover the face structure. Jebara et al. [25] relied in fixed feature points (eyes, nose tip), while [63] tracked only points with high Hessian values. Also, [25] tracked 2D features in 3D by deforming them, while [63] relied on direct comparison of a 3D model to the image. Methods have been proposed in [61] and [64] to solve the varying appearance (both geometry and photometry) problem in tracking. Some of the newest model-based tracking methods calculate the 3D motions and deformations directly from image intensities [65], thus eliminating the information-lossy intermediate representations.

(3) Face Modeling.

Modeling of faces includes 3D shape modeling and texture modeling. 3D models of faces have been employed in the graphics, animation, and model-based image compression literature. More complicated models are used in applications such as forensic face reconstruction from partial information.

In computer vision, one of the most widely used methods of estimating 3D shape from a video sequence is SfM, which estimates the 3D depths of interesting points. The unconstrained SfM problem has been approached in two ways. In the differential approach, one computes some type of flow field (optical, image, or normal) and uses it to estimate the depths of visible points. The difficulty in this approach is reliable computation of the flow field. In the discrete approach, a set of features such as points, edges, corners, lines, or contours are tracked over a sequence of frames, and the depths of these features are computed. To overcome the difficulty of feature tracking, bundle adjustment [66] can be used to obtain better and more robust results.

Recently, multiview based 2D methods have gained popularity. In [55], a model consisted of a sparse 3D shape model learned from 2D images labeled with pose and landmarks, a shape-and-pose-free texture model, and an affine geometrical model. An alternative approach is to use 3D models such as the deformable model of [67] or the linear 3D object class model of [73]. (In [73] a morphable 3D face model consisting of shape and texture was directly matched to single/multiple input images; as a consequence, head orientation, illumination conditions, and other parameters could be free variables subject to optimization.) In [73], real-time 3D modeling and tracking of faces was described; a generic 3D head model was aligned to match frontal views of the face in a video sequence.

Approach	Representative work
Still-image methods	Basic methods [10], [81], [5], [82], [83], [68], [84]; Tracking-enhanced [15], [69], [53], [70].
Multimodal methods	Video- and audio-based [71], [52].
Spatiotemporal methods	Feature trajectory-based [56], [57];

	Video-to video methods [72].
--	------------------------------

Table 2.1. Categorization of Video-Based Face Recognition Techniques

2.4.2.2 Video-Based Face Recognition

Face recognition from video sequences originated from still-image-based techniques (Table 2.2). That is, the system automatically detects and segments the face from the video frame, and then applies still-image face recognition techniques. Many methods belong to this category: eigenfaces [10], probabilistic eigenfaces [5], the EBGM (elastic bunch graph matching) method [82], [84], and the PDBNN (probabilistic decision-based neural network) method [81]. An improvement over these methods is to apply tracking; this can help in recognition, in that a virtual frontal view can be synthesized via pose and depth estimation from video. Due to the abundance of frames in a video, another way to improve the recognition rate is the use of “voting” based on the recognition results from each frame. The voting can be deterministic, but probabilistic voting is better in general [53], [73]. One drawback of such voting schemes is the expense of computing the deterministic/probabilistic results for each frame.

The next phase of video-based face recognition will be the use of multimodal cues. Since humans routinely use multiple cues to recognize identities, it is expected that a multimodal system will do better than systems based on faces only. More importantly, using multimodal cues offers a comprehensive solution to the task of identification that might not be achievable by using face images alone. For example, in a totally noncooperative environment, such as a robbery, the face of the robber is typically covered, and the only way to perform faceless identification might be to analyze body motion characteristics [74]. Excluding fingerprints, face and voice are the most frequently used cues for identification. They have been used in many multimodal systems [71], [52].

A third phase of video face recognition has started. These methods [56], [57] coherently exploit both spatial information (in each frame) and temporal information (such as the trajectories of facial features). A big difference between these methods and the probabilistic voting methods [73] is the use of representations of joint temporal and spatial information for identification.

In [68], a fully automatic person authentication system was described which included video break, face detection, and authentication modules. Video skimming was used to reduce the number of frames to be processed. The video break module, corresponding to key frame detection based on object motion, consisted of two units. The first unit implemented a simple optical flow method; it was used when the image SNR level was low. When the SNR level was high, simple pair-wise frame differencing was used to detect the moving object. The face detection module consisted of three units: face localization using analysis of projections along the x- and y-axes; face region labeling using a decision tree learned from positive and negative examples taken from 12 images each consisting of 2759 windows of size 8 x 8; and face normalization based on the numbers of face region labels. The normalized face images were then used for authentication, using an RBF (radial basis function) network. This system was tested on three image sequences; the first was taken indoors with one subject present, the second was taken outdoors with two subjects, and the third was taken outdoors with one subject under stormy conditions. Perfect results were reported on all three sequences, as verified against a database of 20 still face images.

An access control system based on person authentication was described in [69]. The system combined two complementary visual cues: motion and facial appearance. In order to reliably detect significant motion, spatiotemporal zero crossings computed from six consecutive frames were used. These motions were grouped into moving objects

using a clustering algorithm, and Kalman filters were employed to track the grouped objects. An appearance-based face detection scheme using RBF networks (similar to that discussed in [75]) was used to confirm the presence of a person. The face detection scheme was “bootstrapped” using motion and object detection to provide an approximate head region. Face tracking based on the RBF network was used to provide feedback to the motion clustering process to help deal with occlusions. Good tracking results were demonstrated. In [53], this work was extended to person authentication using PCA or LDA. The authors argued that recognition based on selected frames is not adequate since important information is discarded. Instead, they proposed a probabilistic voting scheme; that is, face identification was carried out continuously. Though they gave examples demonstrating improved performance in identifying 8 or 15 people by using sequences, no performance statistics were reported.

An appearance model based method for video tracking and enhancing identification was proposed in [15] the appearance model is a combination of the active shape model (ASM) [76] and the shape-free texture model after warping the face into a mean shape. Unlike [85], which used the two models separately, the authors used a combined set of parameters for both models. The main contribution was the decomposition of the combined model parameters into an identity subspace and an orthogonal residual subspace using linear discriminant analysis. The residual subspace would ideally contain intraperson variations caused by pose, lighting, and expression. In addition, they pointed out that optimal separation of identity and residue is class-specific. For example, the appearance change of a person’s nose depends on its length, which is a person-specific quantity. To correct this class-specific information, a sequence of images of the same class was used. Specifically, a linear mapping was assumed to capture the relation between the class-specific correction to the identity subspace and the intraperson

variation in the residual subspace. Examples of face tracking and visual enhancement were demonstrated, but no recognition experiments were reported. Though this method is believed to enhance tracking and make it robust against appearance change, it is not clear how efficient it is to learn the class-specific information from a video sequence that does not present much residual variation.

In [67], a system called PersonSpotter was described. This system is able to capture, track, and recognize a person walking toward or passing a stereo CCD camera. It has several modules, including a head tracker, preselector, landmark finder, and identifier. The head tracker determines the image regions that are changing due to object motion based on simple image differences. A stereo algorithm then determines the stereo disparities of these moving pixels. The disparity values are used to compute histograms for image regions. Regions within a certain disparity interval are selected and referred to as *silhouettes*. Two types of detectors, skin color based and convex region based are applied to these silhouette images. The outputs of these detectors are clustered to form regions of interest that usually correspond to heads. To track a head robustly, temporal continuity is exploited in the form of the thresholds used to initiate, track, and delete an object.

To find the face region in an image, the preselector uses a generic sparse graph consisting of 16 nodes learned from eight examples face images. The landmark finder uses a dense graph consisting of 48 nodes learned from 25 example images to find landmarks such as eyes and the nose tip. Finally, an elastic graph matching scheme is employed to identify the face. A recognition rate of about 90% was achieved.

A multimodal person recognition system was described in [52]. This system consists of a face recognition module, a speaker identification module, and a classifier fusion module. It has the following characteristics: (1) the face recognition module can

detect and compensate for pose variations; the speaker identification module can detect and compensate for changes in the auditory background; (2) the most reliable video frames and audio clips are selected for recognition; (3) 3D information about the head obtained through SfM is used to detect the presence of an actual person as opposed to an image of that person.

Two key parts of the face recognition module are face detection/ tracking and eigenface recognition. The face is detected using skin color information using a learned model of a mixture of Gaussians. The facial features are then located using symmetry transforms and image intensity gradients. Correlation-based methods are used to track the feature points. The locations of these features points are used to estimate the pose of the face. This pose estimate and a 3D head model are used to warp the detected face image into a frontal view. For recognition, the feature locations are refined and the face is normalized with eyes and mouth in fixed locations. Images from the face tracker are used to train a frontal eigenspace, and the leading 35 eigenvectors are retained. Face recognition is then performed using a probabilistic coefficient of all images of each person is modeled as a Gaussian distribution.

Finally, the face and speaker recognition modules are combined using a Bayes net. The system was tested in an ATM (Automatic Teller Machine) scenario, in a controlled environment. An ATM session begins when the subject enters the camera's field of view and the system detects his/her face. The system then greets the user and begins the banking transaction, which involves a series of questions by the system and answers by the user. Data for 26 people were collected; the normalized face images were 40 x 80 pixels and the audio was sampled at 16 kHz. These experiments on small databases and well-controlled environments showed that the combination of audio and video improved

performance, and that 100% recognition and verification were achieved when the image/audio clips with highest confidence scores were used.

In [78], a face verification system based on tracking facial features was presented. The basic idea of this approach is to exploit the temporal information available in a video sequence to improve face recognition. First, the feature points defined by Gabor attributes on a regular 2D grid are tracked. Then, the trajectories of these tracked feature points are exploited to identify the person presented in a short video sequence. The proposed tracking-for-verification scheme is different from the pure tracking scheme in that one template face from a database of known persons is selected for tracking. For each template with a specific personal ID, tracking can be performed and trajectories can be obtained. Based on the characteristics of these trajectories, identification can be carried out. According to the authors, the trajectories of the same person are more coherent than those of different persons. Such characteristics can also be observed in the posterior probabilities over time by assuming different classes. In other words, the posterior probabilities for the true hypothesis tend to be higher than those for false hypotheses. This in turn can be used for identification. Testing results on a small database of 19 individuals have suggested that performance is favorable over a frame-to-frame matching and voting scheme, especially in the case of large lighting changes. The testing results are based on comparison with alternative hypotheses.

Some details about the tracking algorithm are as follows [56]. The motion of facial feature points is modeled as a global two-dimensional affine transformation (accounting for head motion) plus a local deformation (accounting for residual motion that is due to inaccuracies in the 2D affine modeling and other factors such as facial expression). The tracking problem has been formulated as a Bayesian inference problem and sequential importance sampling (SIS)[77](one form of SIS is called *Condensation* [78] in the

computer vision literature) proposed as an empirical solution to the inference problem. Since SIS has difficulty in high-dimensional spaces, a reparameterization that captures essentially only the difference was used to facilitate the computation.

A video-based face recognition approach that takes video sequences as input has been developed [72]. Since the detected face might be moving in the video sequence, one has to deal with uncertainty in tracking as well as in recognition. Rather than solving these two uncertainties separately, Zhou et al. [72] performed simultaneous tracking and recognition of human faces from a video sequence.

In still-to-video face recognition, where the gallery consists of still images, a time series state space model is proposed to fuse temporal information in a probe video, which simultaneously characterizes the kinematics and identity using a motion vector and an identity variable, respectively. The joint posterior distribution of the motion vector and the identity variable is first estimated at each time instant and the propagated to the next time instant. Marginalization over the motion vector yields a robust estimate of the posterior distribution of the identity variable and Marginalization over the identity variable yields a robust estimate of the posterior distribution of the motion vector, so that tracking and recognition are handled simultaneously. A computationally efficient sequential importance sampling (SIS) algorithm is used to estimate the posterior distribution. Empirical results demonstrate that, due to the propagation of the identity variable over time, *degeneracy* in the posterior probability of the identity variable is achieved to give improved algorithm. The gallery is generalized to videos in order to realize video-to-video face recognition. An exemplar-based learning strategy is employed to automatically select video representatives from the gallery, serving as mixture centers in an updated likelihood measure. The SIS algorithm is used to approximate the posterior distribution of the motion vector, the identity variable, and the exemplar index. The

marginal distribution of the identity variable produces the recognition result. The model formulation is very general and allows a variety of image representations and transformations.

In [57], a multiview based face recognition system was proposed to recognize faces from video with large pose variations. To address the challenging pose issue, the concept of an *identity surface* that captures joint spatial and temporal information was used. An identity surface is a hypersurface formed by projecting all the images of one individual onto the discriminating feature space parameterized on head pose. To characterize the head pose, two angles, yaw and tilt, are used as basis coordinates in the feature space. The other basis coordinates represent the discriminating feature patterns of faces. Based on recovered pose information, a trajectory of the input feature pattern can be constructed. The trajectories of features from known subjects arranged in the same temporal order can be synthesized on their respective identity surfaces. To recognize a face across views over time, the trajectory for the input face is matched to the trajectories synthesized for the known subjects. This approach can be thought of as a generalized version of face recognition based on single images taken at different poses. Experimental results using twelve training sequences, each containing one subject, and new testing sequences of these subjects was reported. Recognition rates were 100% and 93.9%, using 10 and 2 KDA (kernel discriminant analysis) vectors, respectively.

Other methods have also been used to construct the discriminating basis in the identity surface: kernel discriminant analysis (KDA) [79] was used to compute a nonlinear discriminating basis, and a dynamic face model is used to extract a shape-and-pose-free facial texture pattern. The multiview dynamic face model [55] consists of a sparse Point Distribution Model (PDM) [76], a shape-and-pose-free texture model, and an affine geometrical model. The 3D shape vector of a face is estimated from a set of 2D

face images in different views using landmark points. Then a face image fitted by the shape model is warped to the mean shape in a frontal view, yielding a shape-and-pose-free texture pattern. When part of a face is invisible in an image due to rotation in depth, the facial texture is recovered from the visible side of the face using the bilateral symmetry of faces. To obtain a low-dimensional statistical model, PCA was applied to the 3D shape patterns separately. To further suppress within-class variations, the shape-and-pose-free texture patterns were further projected into a KDA feature space. Finally, the identity surface can be approximated and constructed from discrete samples at fixed poses using a piece-wise planar model.

Face recognition image sequence have a unique advantage over still image approaches: the abundance of temporal information. But, the low quality of video images presents a problem: the loss of spatial information. Thus, using temporal information to compensate for the lost spatial information is vital to build a successful video-based system.



2.5 SUMMARY

Face detection has successfully been applied as prior step for face tracking and recognition in the process of remote surveillance. In this chapter, we have defined what face detection is, we then discussed the state of the art of face detection techniques. Prior to that, we gave an overview of remote surveillance systems. Furthermore, we discussed different methods of face recognition from image sequences. This then motivates and puts us in the position to be able to present the issue of face detection using skin color modeling in the HSV color space in chapter 3.

CHAPTER 3

FACE DETECTION USING HSV SKIN COLOR MODELING

3.1 INTRODUCTION

The sections that follow in this chapter give a detailed description of our proposed solution for face detection using HSV skin colour modeling. This method is used as a prior step for the face tracking system presented in chapter six.

To segment human skin regions from an image, a reliable skin colour model that is adaptable to different skin colours and to different lighting conditions is needed. The basic RGB colour model is not suitable for skin colour characterization because the triplet (R, G, B) represents not only colour but also luminance, which vary across the person's face due to the ambient lighting. One of the most significant parts of this project was to find an appropriate skin color model in order to facilitate real-time face detection, which is adaptable to people of different skin colors and to different lighting conditions. Thus, we have found a skin color model in the HSV color space to be appropriate, because HSV gives the best performance for skin pixel detection [79].

HSV colour model is a nonlinear transformation of the RGB colour space. The HSV model is user-oriented and is based on the artist's notions of tint, shade, and tone,

with independent values for Hue, Saturation, and Value, corresponding, respectively to wavelength, excitation, and brightness. **Figure 3.1** shows the HSV coordinate system as a hexacone:

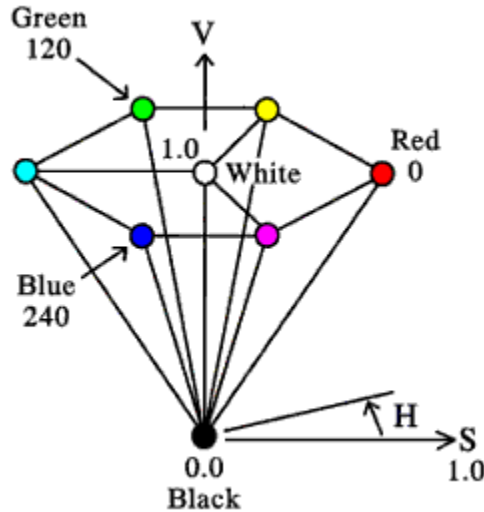


Figure 3.1: The HSV Color Model

Black has HSV = (0, 0, 0). Thus the facial HSV model of Africans is clustered around the origin of the coordinate system. Across the African continent, modest variations occur which do not depart considerably from this value. Most colour pictures are recorded as (R, G, B) triplets. Given a colour defined by (R, G, B) where R, G, and B are normalized to 0.0 to 1.0, an equivalent (H, S, V) colour is determined by the following set of formulas (3.1) and (3.2). Considering MAX to be the maximum of the (R, G, B) values, and MIN the minimum of those values. The formula is then expressed as [86]:

$$H = \begin{cases} 0 + \frac{G - B}{MAX - MIN} \times 60, & \text{if } R = MAX \\ 2 + \frac{B - R}{MAX - MIN} \times 60, & \text{if } G = MAX \\ 4 + \frac{R - G}{MAX - MIN} \times 60, & \text{if } B = MAX \end{cases} \quad (3.1)$$

$$S = \frac{MAX-MIN}{MAX}; \quad V = MAX \quad (3.2)$$

Where H varies from 0.0 to 360, indicating the angle in degrees, S and V vary from 0.0 to 1.0.

In mobile phone photos, the face often dominates the photo in terms of the number of pixels used to represent it compared to the other parts of the body. Therefore, algorithms using face colour features are likely to perform better on handsets.

The proposed and implemented face detection technique uses skin color as the basic feature for face detection. It consists of a statistical skin color modeling in the HSV color space, skin color segmentation and the selection of face regions. The following sections discuss these phases in detail.

3.2 HSV SKIN COLOR MODEL



First of all, we create a skin color model in the HSV color space. To create the skin color model, we cut skin regions from 23 true color images of people of different ethnic backgrounds. Each of these skin color samples is then converted to the HSV color space as shown in **figure 3.2**. The conversion of each color image to HSV space eliminates the luminance component of the RGB space, because lighting effects change the appearance of the skin.

To convert from RGB to HSV color space, the MATLAB function “*rgb2hsv*” is used.

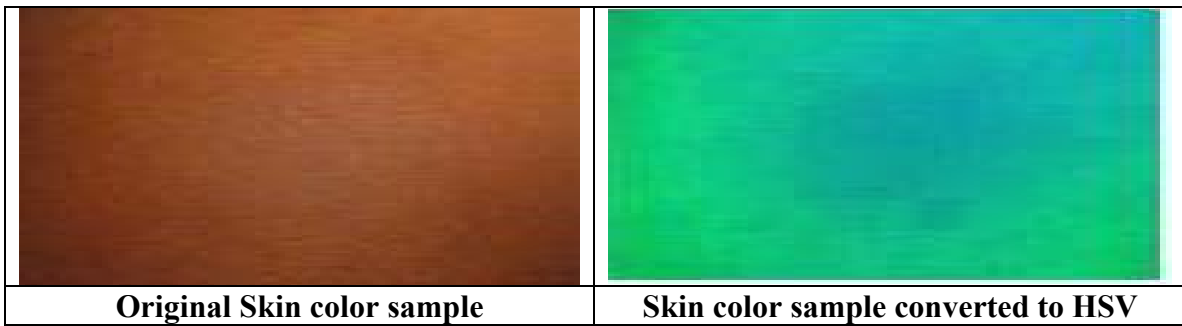


Figure 3.2: Shows the Skin sample in the RGB space and the HSV space

The 3D color histogram illustrated in **figure 3.3**, revealed that the color distributions of skin color of different people are clustered in small area of the H-S color space.

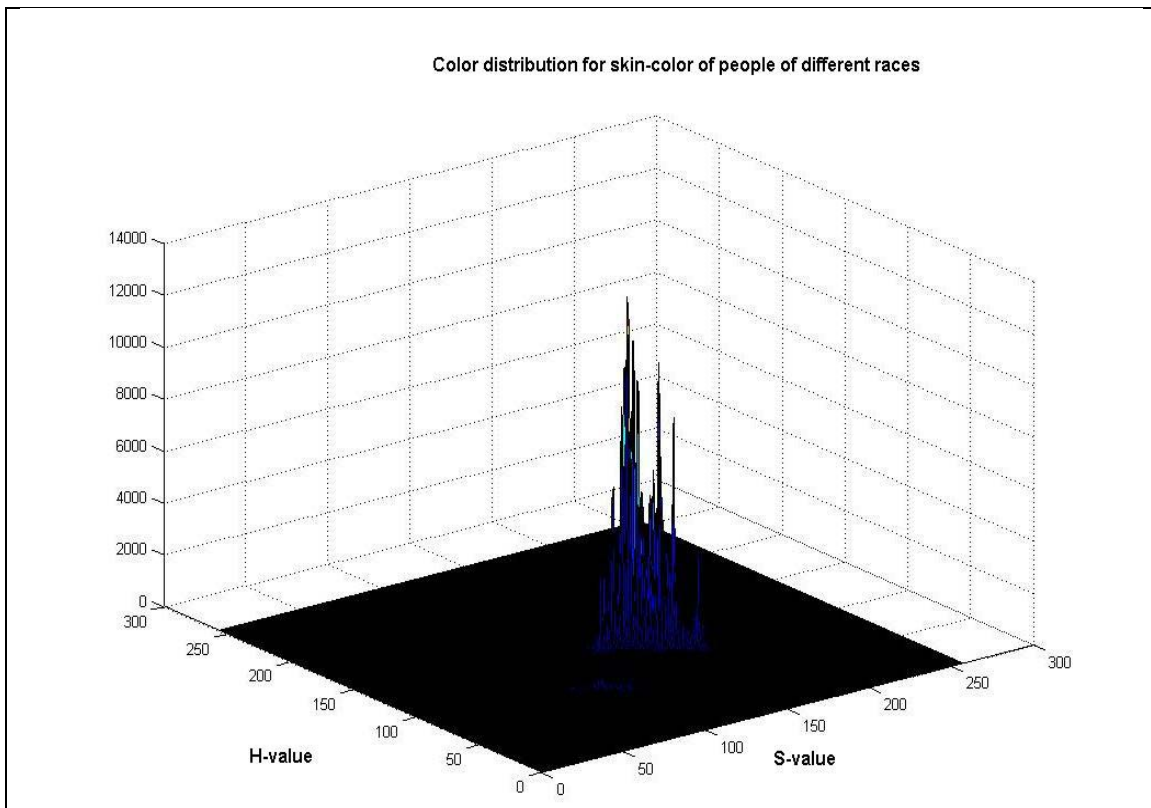


Figure 3.3. Color distribution in the HSV color space for skin color of different people of various ethnic groups

We then build a mean and covariance model of H- and S-values from the HSV² color space:

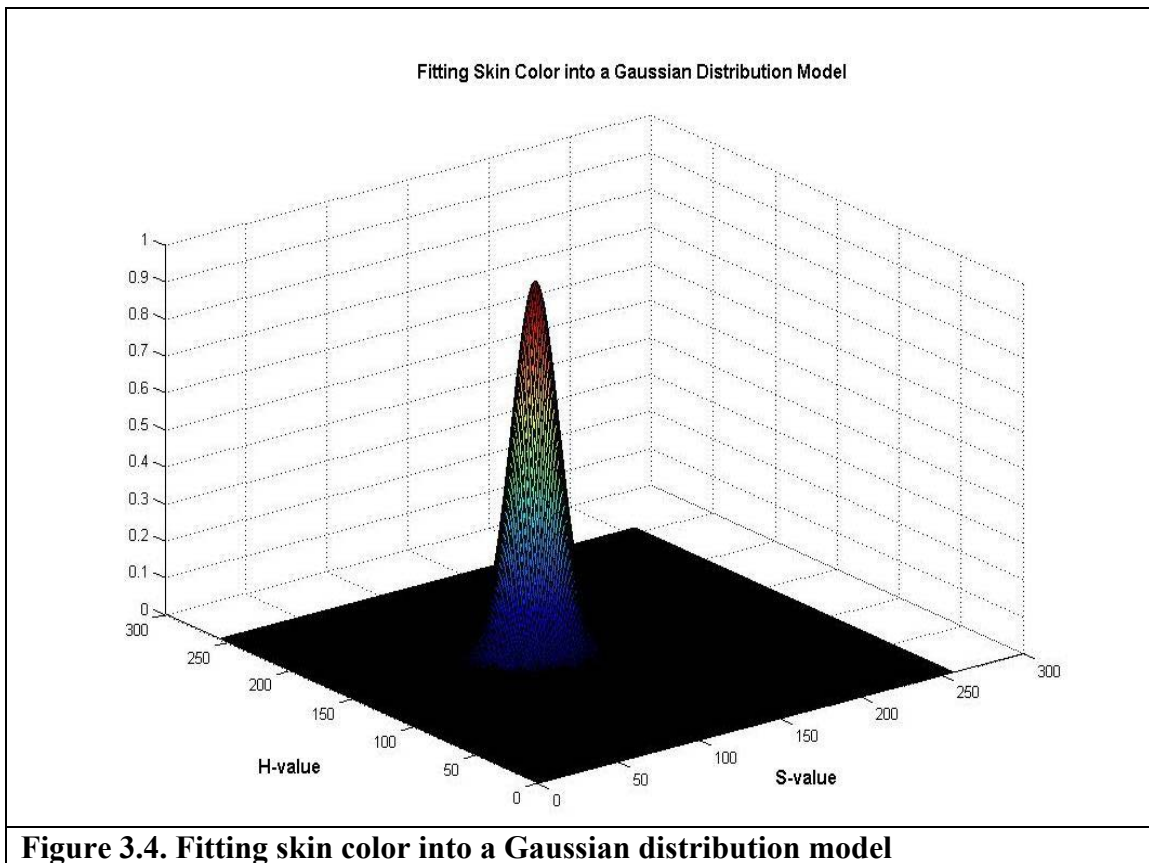
² The values for V were discarded, only took into consideration the H- and S- values.

$$\text{Mean : } m = E\{x\} \quad (3.3)$$

$$\text{Covariance : } C = E\{(x - m)(x - m)^T\} \quad (3.4)$$

Where $x = (H \ S)^T$

With the mean and covariance values, the skin color model can be fitted into a Gaussian model by $N(m, C)$, by plotting H against S as illustrated in **figure 3.4**.



The mean vector is often referred to as the centroid and the covariance matrix as the dispersion matrix. Figure 3.4 is an illustration of a Gaussian distribution $N(m, C)$ model fitted by our data.

3.3 SKIN COLOR SEGMENTATION

With the skin color model, the segmentation process can begin. The first step is to modify the original image into the color space in which we want to detect a face. Thus, to eliminate the luminance component of the original image, in the RGB color space, we convert the image into the HSV color space as indicated previously (**Figure 3.5**).

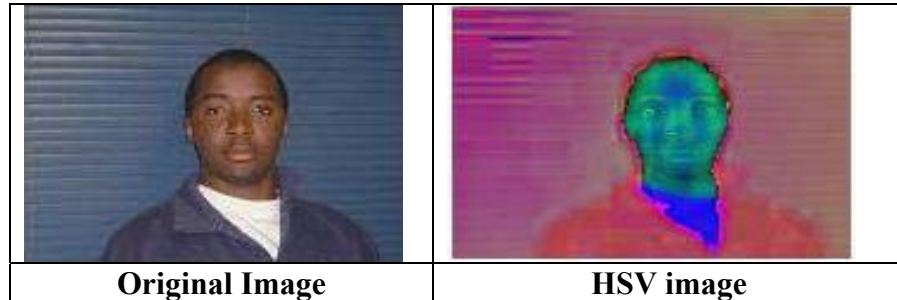


Figure 3.5: shows the RGB image and the equivalent HSV image

Then (**Figure 3.6**) we segment the original image to find regions that are most likely a skin region using the skin Gaussian model and the following expression:



$$P(H, S) = \exp\left[-0.5(x - m)^T C^{-1}(x - m)\right] \quad (3.5)$$

Where $x = (H \ S)^T$

The above expression determines the probability of one pixel being a skin region based on the skin color model. This probability is given on a grayscale skin probability image based on the gray level.



Original Image	Skin-likelihood Image
-----------------------	------------------------------

Figure 3.6: shows the original image next to the skin-likelihood image

Further segmentation is then done (**Figure 3.7**) to threshold the grayscale skin likelihood image into a binary image. Since skin colors vary between people, an adaptive threshold process to find the optimal threshold value is used. There is no mathematical expression for this adaptive threshold, the values were chosen in an ad hoc manner. If the threshold value is too low, the amount of segmented skin regions increases. Based on this assumption, an adaptive threshold is created. This method starts the threshold value at 0.65 and decrements by factors of 0.1 until it gets to 0.05. The program determines the optimal threshold value by finding the point when the change in the number of segmented regions is minimum. After the optimal threshold value is determined, the grayscale image is then converted to binary. In the binary image, the white regions show the skin regions and are labeled as 1 and the black regions show the non-skin regions and are labeled as 0. Each of these skin regions is assigned an integer value they are tested individually to see if it represents a face.

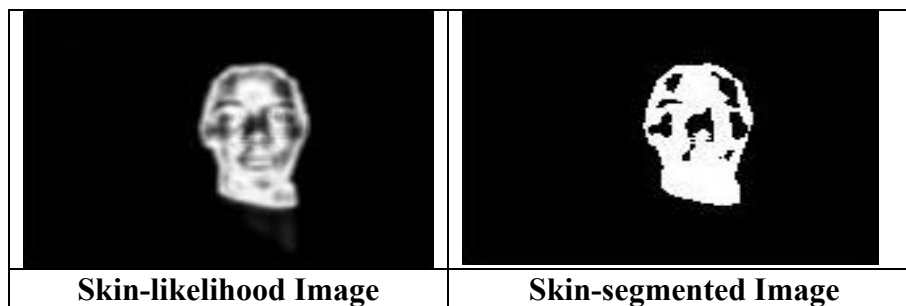


Figure 3.7: Shows the skin-likelihood image and the skin-segmented image

3.4 SELECTION OF FACE REGIONS

Using the results from the previous section, we proceed to determine which regions can possibly determine a human face. We look for a closed white (skin) region that has

one or more holes (black region) inside it. It is the same as look for a black region that is bounded with a white region. The algorithm checks for every pixel of the black region, if we move from the pixel to the left, to the right, up and down, we should find four pixels that are part of the same white region. In other words, the white region has a hole inside it. The black holes inside white regions result from facial features such as nose, mouth, and eyes. Thus, to determine the number of holes in a white region we use the following expression:

$$H = 1 - E \quad (3.6)$$

Where H is the number of holes in a region and E is the Euler number. We use 1 because we are analyzing only one segmented region at a time.

If an area with at least one hole in it is found, we continue to find other characteristics about the region before concluding that this area constitutes a face.

The center of mass of the area with at least one hole in it calculated by:

$$\bar{X} = \frac{1}{A} \sum_{i=1}^n \sum_{j=1}^m jB[i, j] \quad (3.7)$$

$$\bar{Y} = \frac{1}{A} \sum_{i=1}^n \sum_{j=1}^m iB[i, j] \quad (3.8)$$

Where B is the matrix of size [n x m] representation of the region, and A is the area in pixels of the region.

In order to include faces with an inclination, with the intention of maximizing the detection rate, we compute the angle of inclination of the face:

$$\theta = \frac{1}{2} a \tan \frac{b}{a - c} \quad (3.9)$$

Where:

$$a = \sum_{i=1}^n \sum_{j=1}^m (x'_{ij})^2 B[i, j] \quad (3.10)$$

$$b = 2 \sum_{i=1}^n \sum_{j=1}^m x'_{ij} x^i_{ij} B[i, j] \quad (3.11)$$

$$c = \sum_{i=1}^n \sum_{j=1}^m (y'_{ij})^2 B[i, j] \quad (3.12)$$



and:

$$x' = X - \bar{X} \quad (3.13)$$

$$y' = Y - \bar{Y} \quad (3.14)$$

After determining the center and the angle of the white region with at least one hole in it, we now calculate the width and the height of that region to improve the decision

process. This includes moving one pointer from the left, one from the right, top and bottom of the image. We compute the coordinate of 4 boundary points, and calculate the height by subtracting the bottom and top values and the width by subtracting the right and the left values.

Another parameter that is also useful to determine if the segmented area is a face is the ratio of height to width. The height to width ratio of a human face is approximately 1. In order to improve the detection rate, we found experimentally that 0.8 is a good minimum value. Values below 0.8 are not helpful to conclude that the region is a face, because human faces are oriented vertically. We determined a good upper limit to be 1.9. However, there are situations where we have indeed human faces but the upper limit of the ratio is higher. This happens when the neck of the person is uncovered. Therefore, to account for this we set the ratio to be 1.9 and we eliminate the region below.



3.5 GENERATION OF AVERAGE FACE MODEL

A 2-Dimensional template matching algorithm was used to build a template face which is used to take the final decision of determining if the skin region represents a face. The template face grasps as much as possible the common features of human face, but is not dependent on the background and individual characteristics of the face. The average face template is generated by enclosing the eye brows and the upper lips, as illustrated in **figure 3.8 (a)**. This template is chosen by averaging 16 frontal faces of males and females as shown in **figure 3.8 (c)**. **Figure 3.8 (b)** illustrates the final template face (model) used to verify the existence of faces in skin regions.

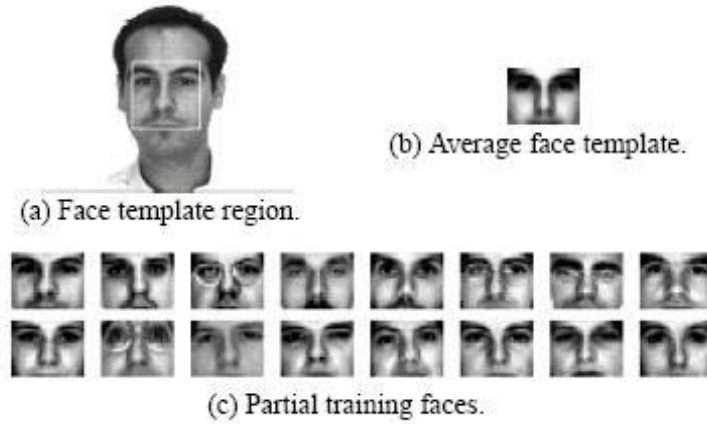


Figure 3.8: Generation of Average Face Template (model)

The template face image is used to fill the area of the image corresponding to the skin region. This process is done by resizing and positioning the template face according to the skin region's characteristics. In other words, the template frontal face is positioned and rotated in the same coordinate as the image corresponding to the skin region. Thus, the width and height values are used to resize the frontal face model into these dimensions. The value of the angle of inclination is used to rotate the resized template face into the same direction as the skin region. The center of mass is used to place the template face exactly at the center of the skin segmented region.

A correlation, which computes the two dimensional correlation coefficients between two matrices, is used to determine how well the template face fits into the skin region. This function is the implementation of the following algorithm:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\left[\sum_m \sum_n (A_{mn} - \bar{A})^2 \right] \left[\sum_m \sum_n (B_{mn} - \bar{B})^2 \right]} \quad (3.15)$$

Where A_m and B_m are matrices of the same size, $\bar{A} = \text{means2}(A_m)$, and $\bar{B} = \text{means2}(B_m)$. The return value r is scalar double.

A good threshold value for classification of a skin region as a face is if the resulting autocorrelation value is greater than 0.6.

Once every region has been successfully evaluated, the original color image is displayed (**Figure 3.9**) with rectangles placed around the faces in the image. This is done by obtaining the coordinates of the part of the image that has the template face (model). With this coordinates, we draw a rectangle in the original color image.

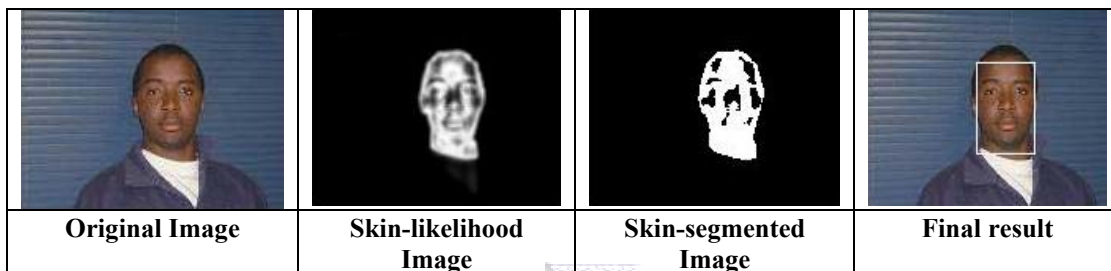


Figure 3.9: General Overview of the described system

3.6 EXPERIMENTAL RESULTS

The system was tested on a set of 51 images. Set 1 is Caucasians, Set 2 is Africans, and Set 3 is Asians, both males and females. This selection of various ethnic groups in the test set was to ensure the robustness of the system. All the images include varying illumination conditions and varying background conditions. **Figure 3.10** illustrates the system's face detection capability using the HSV color model, from (a) to (d), the first images are the original test images that are being tested for instances of a face, next are the skin likelihood images, then are the segmented binary images, the last images are the original images with a box drawn around the face. The program worked quite well in example 1 of **figure 3.10**, where in the image we have a face of dark skin color. The system successfully recognized the single instance of a face in the image, but it also included the area of the neck. This happened during the labeling of each area, the neck

was included with the face area as one labeled region. In example 2, where we have an image of a Caucasian lady, the program successfully detected the single instance of the face in the image.

Figure 3.10: Some of the tests using the HSV Color Space

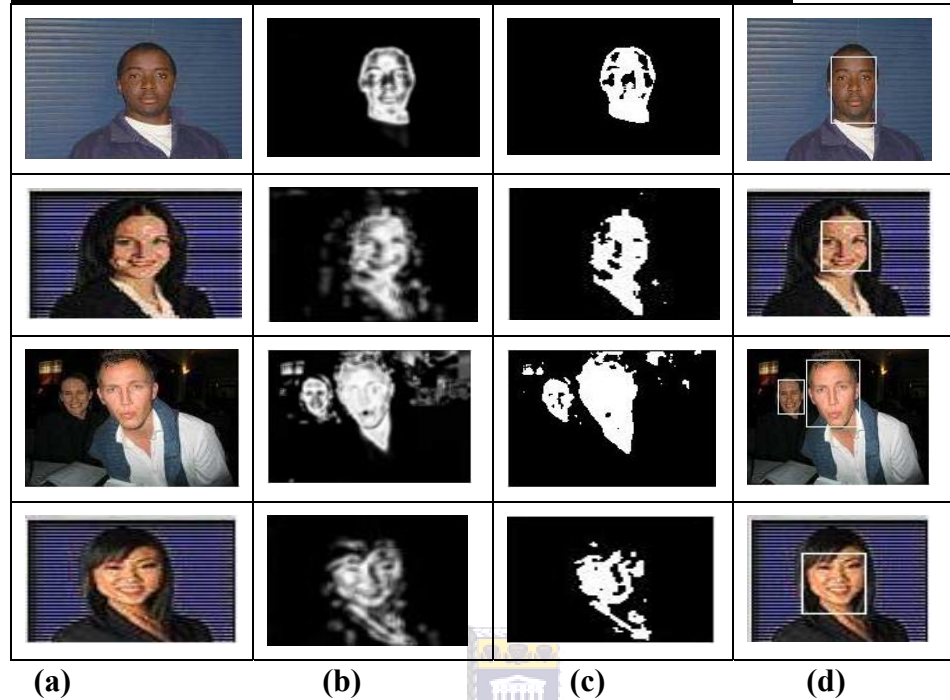


Figure 3.10: (a) The original color images; (b) Skin-likeness images; (c) skin-segmented images; (d) final results

For example 3 of **figure 3.10**, it is a demonstration of how multiple instances of faces can be detected. We have in the image a Caucasian couple, and the program successfully recognized the two instances of the faces in the image including the faded face at the background. In the last example we have an image of an Asian (Chinese) lady with her hair covering part of her forehead. The program worked relatively well, the hair in her forehead did not affect the detection results, and however, part of the area of her neck was included in the face region, this obviously happened during the labeling process as described for example 1. Otherwise, the system successfully recognized the single instance of a face in the image.

Table 3.1 gives the face detection results for the HSV technique, 59 faces are correctly detected, 7 are missed and 2 faces are falsely detected. The likely reasons for the faces that are missed result from bad lighting conditions and background distractors.

Set of color Images	Correctly detected	Missed	Falsely detected
Set 1	25	3	0
Set 2	19	2	0
Set 3	15	2	2
Total	59	7	2

Table 3.1. Performance of algorithm with HSV model

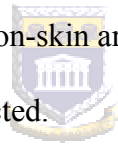
Another reason for the undetected face is due to the very different color of skin or face profiles found across several subjects.

The proposed method is not totally accurate; therefore, it was necessary to run a set of tests to measure the accuracy of the program. During the testing we encountered that the optimal threshold values varied from 0.1 to 0.5. The optimal threshold values with more detection of faces were 0.5 followed by 0.4. We also observed experimentally that minimum value for the height to width ratio was about 0.8, and the maximum value was around 1.9. Cases with ratio below 0.8 were usually a miss, and with values above 1.9 few instances of faces were detected but they included large parts of the neck and the hair. For the height to width ratio on the range from 0.8 through to 1.9 the least amount of non-faces is found and the highest amount of faces was detected. Most of the non-face regions were eliminated by the height to width ratio test.

Through these tests we found out that the best correlation value was about 0.6, because the highest percentage of faces is detected and the least amount of non-faces is found at this value. Moreover, the correlation values incremented from 0.1 to 0.8.

We ran tests on 51 distinct images with a total of 66 faces in order to determine the percentage of faces detected by our approach. With the best value of 0.6 for the correlation and the ratio of height to width on the range from 0.8 to 1.9, the percentage of faces included on the rectangle of the output image was around 89 % (59 faces). The face is considered detected if it is inside the final result box. We encountered one image that included two faces in a box, but counted these two faces as being detected. In some images the final result box only contained part of the persons face, for example the chin is cut out but the other facial features are included.

To measure the accuracy of the algorithm we noticed that in very few cases non-face areas were also detected. To be precise, in two instances some non-face areas were falsely detected as faces. It is cumbersome to give results regarding the non-face images because these results were reliant on the specific images. However, it is necessary to have a high percentage of faces detected with non-skin areas occasionally being detected rather than faces in images not being detected.



3.6.1 Convolutional Neural Network

Based on the assumption that using skin color for the detection of human faces in a real-time system has been proved to [87], [88] have several advantages compared to other methods, because processing for color information is much faster than processing other facial features and under constant lightning conditions, color is nearly invariant against changes in size, orientation and partial occlusions of the face we chose skin color as the basic feature for our face detection technique. Another reason for this choice stems from the low complexity of skin color approaches.

Nevertheless, Garcia and Delakis [89] proposed a convolutional neural network approach designed to recognize strongly variable face patterns directly from pixel images

with no preprocessing, by automatically synthesizing its own set of feature extractors from a large training set of faces.

The convolutional neural network consists of a set of three different kinds of layers. Layer C_i is connected to the retina, receiving the image area to classify as face or non-face and it has a certain number of planes, and they are the so called convolutional layers. Layer S_i is the layer that performs local averaging and subsampling operation. This subsampling reduces the dimensionality of the input by 2 and increases the degrees of invariance to translation, rotation, scale, and deformation of the face patterns.

Layers N_i contain simple sigmoid neurons. The role of these layers is to perform classification, after feature extraction and input dimensionality reduction are performed.

For the training of their large set of faces they used the classical backpropagation algorithm with momentum modified for use on convolutional networks. They built the training set by manually cropping 2146 highly variable face areas in a large collection of images. They chose an input window for the central part of the face with dimension around 20x20, excluding the border of the face and any background. This gives the network some additional information, which is helpful in characterizing the face pattern and canceling some border effects that may arise in the convolutions. The network has 897 trainable parameters, despite the 127,093 connections it uses. Their final training set is of 12,967 faces.

They also collect non-face examples via an iterative bootstrapping procedure. They first build an initial training set of non face examples by producing random images. The network is then trained with face and non face examples. This process is done iteratively to gather false alarms and they obtain a set of 15,000 false examples.

We evaluated this method using our test data set, which contains images that present large variability in size, illumination, facial expression, orientation, and partial occlusions. We present some results of this method on our data set.

Test Image	Results using Neural Networks approach	Results using HSV model approach
		
		
		
		
		

Figure 3.11: Test results of a NN approach and HSV model

The above test results show that the convolutional Neural Network approach does not detect small faces; it also fails to detect faces that are more than 20 degrees rotated, as illustrated respectively on example 2, and 3. In example 4, the test image contains a face of a little girl with drawings and paintings in it, the feature had many features to extract

besides the normal features that a human face contains, which is not ordinary for a face and therefore it concluded that it has not found an instance of a face in that image.

3.6.2 Computing Time

Our data shows that the runtime for this HSV approach scales linearly with the number of people in the test image (for N people, $runtime \approx (N + 1) \times 400ms$). For real-time performance, a face detection domain is limited to 1 person on an image with face size between 40×40 to 70×70 pixels (around 0.8 seconds) when running the detection code on handheld computers.

The memory footprint of the face detection algorithm is $(3 + N \times 0.006)$ MB for N people. This scales to just over 3MB for 20 people, which is reasonable for handheld devices.

The runtime during the training phase for our 23 skin samples to build the skin color model is around 5 seconds. And we did not need to train non-skin samples, yet the performance is 89 %. And the memory footprint for the training phase of the 23 skin samples to build the skin color model is 3.138MB.

The code size for this method is around 20.727 KB.

Our data shows that the runtime for the Convolutional Neural Network approach scales linearly with the number of people (for N people, $runtime \approx (N + 1) \times 400ms$). And the number of compressed images required to be stored in memory is very large (12967 faces and 15000 false examples). The runtime for the training phase of the 12967 faces and the 15000 false examples would require 93.2 hours of computing time, just over 3 days.

The memory footprint of the face detection algorithm is $(3 + N \times 0.006)$ MB for N people. This scales to just over 173.8MB for 12967 faces and the 15000 false examples.

The code size for this method including all the data is around 1.46 MB.

3.7 DISCUSSION

This proposed and implemented program is fast and reliable; it successfully detects instances of front faces with or without a slight angle of inclination in color images. The algorithm is used to segment and detect multiple instances of faces in images. The runtime behavior leads to the conclusion that it can be used in real time video applications. From our experimentation it was 89 % precise. All the images included varying illumination conditions and varying background conditions. The program also works very well with images where the face has shading in it. But for images where the faces are very bright, the segmentation process was inadequate. In images with background that has color similar to that of the skin the program has some problems too.

Considering that the code thresholds the areas and then labels each area, neighboring skin likely regions are occasionally combined, because the program labels two skin likely regions as one region, which often changes the shape of the region and might miss the actual object. For example, we observed during our experimentation that in one image two faces were detected as one face on the final result box, because the shape of the segmented regions of the two faces was changed due to the fact that they were standing very close to one another, and it changed the correlation value also, and the template image was very large and placed on the center of the combined region. In this case a big final result box is drawn including the face and part of the background. The same happened in images where the background or part of it contained colors that were similar to skin and those parts of the background were very close to the face. Different solutions such as a neural network (NN) program, principal component analysis (PCA) or support vector machine (SVM) need to be researched and tested in order to correct this kind of problems.

A standard deviation of the pixel gray levels for the face candidates could be used to remove non-faces caused by uniform skin-color-like regions, such as lights, clothes, walls. Non-face templates, such as hand, arms, and neck templates could be implemented to remove other body parts. Additional face templates could be used to detect the missing faces.

This method would work best for taking the first step in face recognition for remote surveillance with or without mobile phones or other applications of face recognition.

The Convolutional Neural Networks method may be combined with optical preprocessing to obtain very fast face detection. However, the training process is very long and very much resource intensive. Thus, the training phase would probably not do well on a hand-held device, but the operational portion is fairly trivial by modern standards and their online application is computationally easy, thus, it is worth investigating its implementation on handheld devices.

The authors reported that this approach is 20 times faster than other approaches which require a dense scanning of the input image at all scales and positions. It processes a 352x288 image in less than 4 seconds using a PC (933 MHz processor and 256M of memory). This indicates that this is almost real-time on a PC, but not adequate for handheld devices because they have an order of magnitude lower than PCs in terms of processing speed and memory. Even by decreasing the dimension of the input image the detection rate would still be far from the detection rate of 25 frames per second which is the threshold to realize real time computing, however, with further preprocessing this could be achieved.

Also for the skin color method we only needed a small number of skin samples (23 in total), which is enough to meet our requirements, and we do not need non-skin samples. This makes the processing time to be fairly small as compared to the

convolutional neural network approach; moreover the complexity of implementation is also much desirable for the skin color method than the neural networks. Therefore, this is the reason why we chose to implement the skin color technique for face detection.

In conclusion, even though the training process of a Neural Network is long and takes a lot of computing power, and also the fact that the implementation of a Neural Network approach is very cumbersome, we believe that it could also be implemented on handheld devices, on the condition that the training process is done offline on a PC, and that it does not require preprocessing during their online application.

3.8 SUMMARY

A detailed description of the low complexity face detection program using skin color model in the HSV color space was given in this chapter. We have presented the findings of our experiments, which enabled us to deduce a number of conclusions based on the HSV skin color modeling to the task of face detection. We also used our test data set to evaluate the performance of a Convolutional Neural Network approach; we presented some experimental results and found the skin color method using HSV color space to be suitable for the face detection phase. In the next chapter we then present another face detection approach using skin color modeling in the YCbCr color space, we then perform a comparison between the two approaches.

CHAPTER 4

FACE DETECTION USING YCbCr SKIN COLOR MODELING

4.1 INTRODUCTION

This chapter identifies the YCbCr color space as another adequate color space to build a skin color model to address the problem of face detection using low complexity algorithms. This skin color model is used to determine the likelihood of the pixel values and segment the image into skin regions and non-skin regions, then evaluate the skin regions individually for instances of faces. We also perform a very concise comparison between the method described in chapter three and this method.

The RGB color space is not the most efficient way for skin color characterization, as it is particularly susceptible to color changes due to quantization, but also because of the luminance component that makes the RGB color space sensitive to lighting, and because of the noise that it contains. Therefore we convert the color images from the RGB components into YCbCr color space. YCbCr is a colour space used in video systems. It is often confused with the YUV color space, and sometimes the terms YCbCr and YUV are used interchangeably, leading to confusion. The YCbCr consists of the luminance (grayscale), the Y-value, and two chroma (color) components, Cb-value and Cr-value. Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value.

YCbCr data can be double precision, but the color space is particularly well suited to `uint8` data. For `uint8` images, the data range for Y is [16, 235], and the range for Cb and Cr is [16, 240]. YCbCr leaves room at the top and bottom of the full `uint8` range so that additional (nonimage) information can be included in a video stream.

YCbCr is converted from an original RGB source image as follows:

$$Y = 0.299R + 0.587G + 0.11B \quad (4.1)$$

$$\begin{aligned}
 Cb &= 0.5 + \frac{(B - Y)}{1.772} \\
 &= -0.168736R - 0.331264G + 0.5B + 0.5 \quad (4.2)
 \end{aligned}$$

$$\begin{aligned}
 Cr &= 0.5 + \frac{(R - Y)}{1.402} \\
 &= 0.5R - 0.418688G - 0.081312B + 0.5 \quad (4.3)
 \end{aligned}$$

These equations transform RGB in [0, 1] to YCbCr in [0, 255].

4.2 YCbCr SKIN COLOR MODEL

In this section, we will describe a model of skin color in the chromatic color space for human skin segmentation. To create this model, we created 24 skin samples from true color images of people of all races and gender. Each of this skin samples is converted into chromatic color (“pure color”) space as illustrated in **figure 4.1**.

Considering that the lighting effects (luminance component of the RGB space) can change the appearance of the skin, we remove the luminance³ from the color representation in the chromatic color space by simply computing:

$$r = \frac{R}{R + G + B} \quad (4.4)$$

$$g = \frac{G}{R + G + B} \quad (4.5)$$

The normalized blue color is redundant because $r + g + b = 1$.

³ Y-value (Luminance) of the YCbCr is discarded; we use Cb- and Cr- value only.

To convert the 20 skin samples from the RGB color space into the chromatic color space, the “*rgb2ycbcr*” function in MATLAB is used. Thus, in **figure 4.1** we show the original skin sample image, and the original image converted into the chromatic color space.

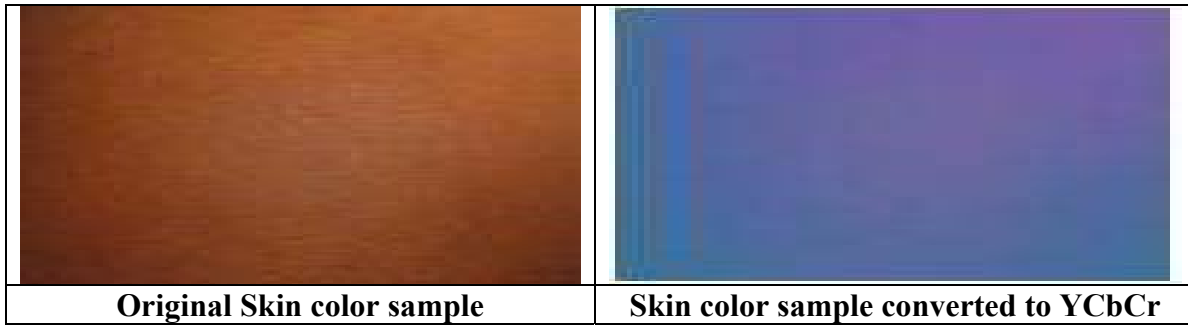
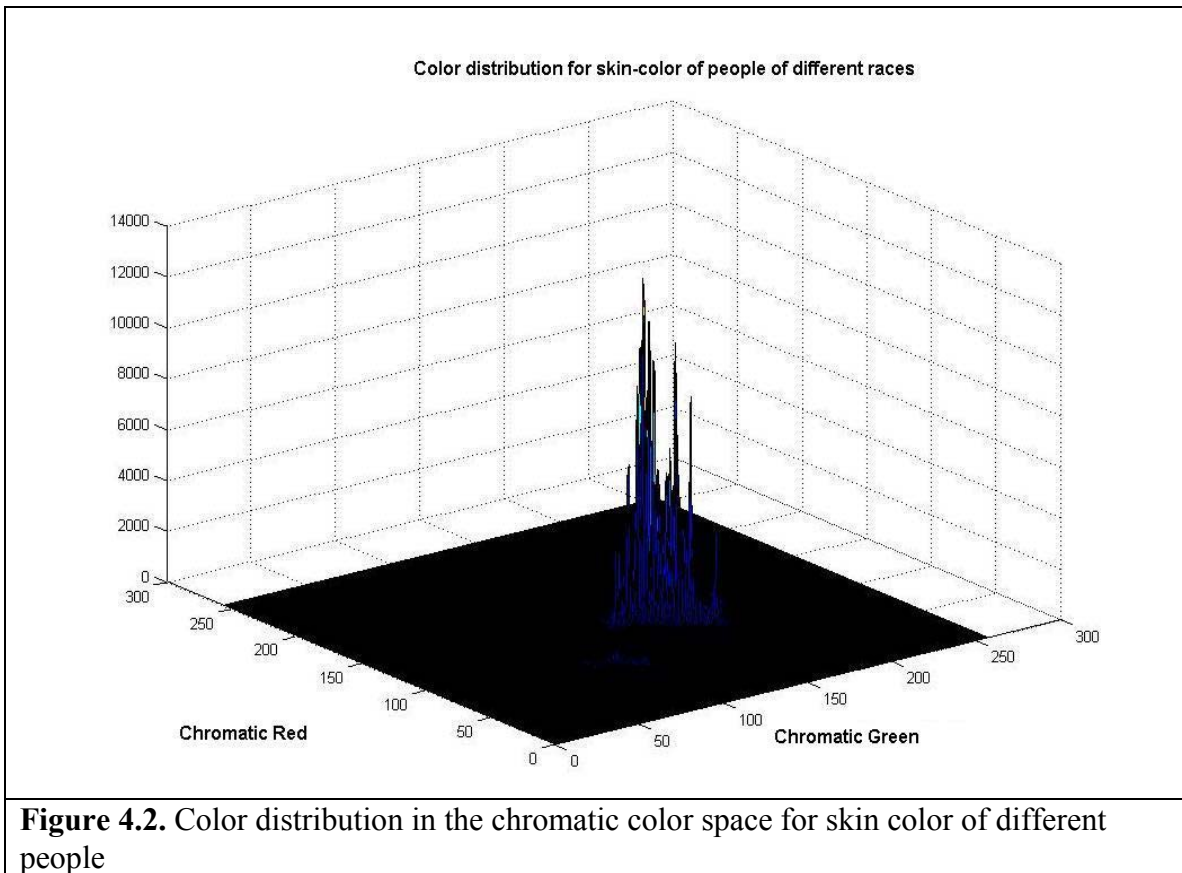


Figure 4.1: Shows the Skin sample in the RGB space and the YCbCr space

Besides the lighting effects (luminance), the RGB color space also has noise in it, therefore each of the rectangles with the skin samples is low-pass filtered to remove the effects of noise. Then we counted the normalized values of red and green color for each pixel of the filtered samples (**equations 4, 5**). The impulse response of the low-pass filter is determined by:

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (4.6)$$

As illustrated in **figure 4.2** the distribution of skin-color of different people is clustered in a small region of the chromatic color space and can be represented by a Gaussian Model.



Gaussian model $N(m, C)$ is a kind of normal statistical model that is estimated with parameters – mean vector and covariance matrix [90]:

$$\text{Mean : } m = E\{x\} \quad (4.7)$$

$$\text{Covariance : } C = E\{(x - m)(x - m)^T\} \quad (4.8)$$

Where $x = (r \ g)^T$.

Figure 4.3 shows a two-dimensional (2-D) Gaussian distribution plot of the mean vector and the covariance matrix of the r and g values.

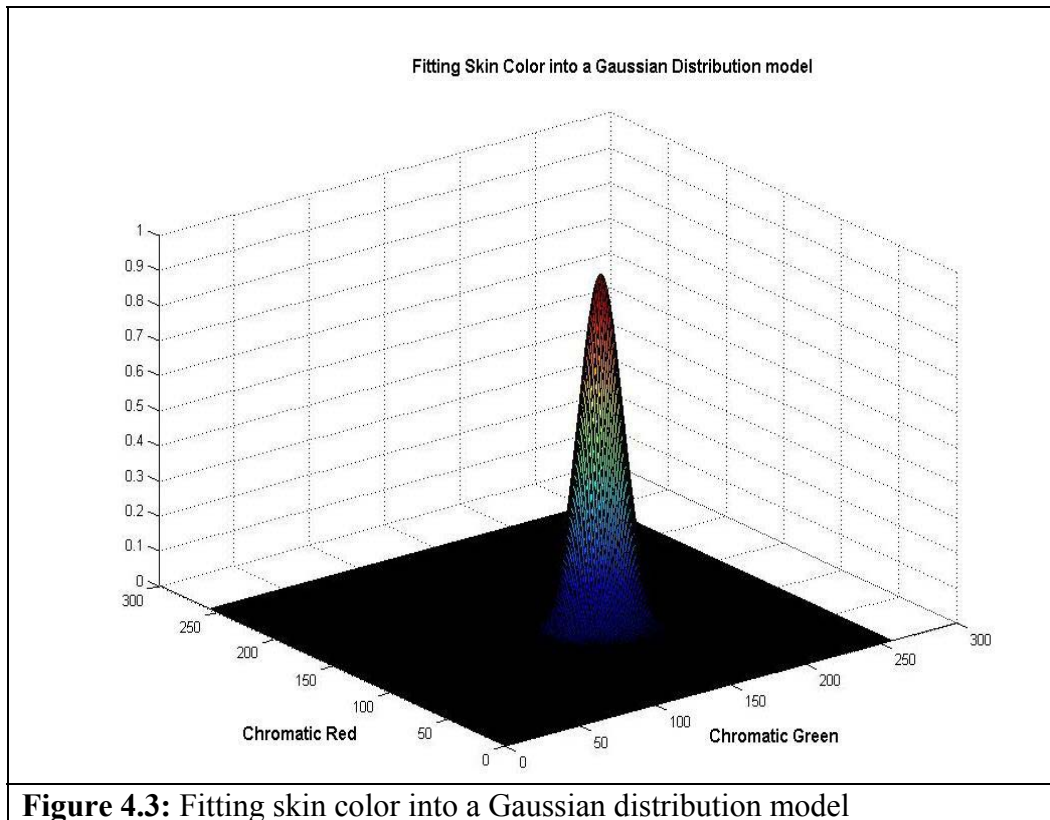


Figure 4.3: Fitting skin color into a Gaussian distribution model

The Gaussian fitted skin color model is used to obtain the likelihood of skin for any pixel of an image. This skin color model transforms a color image into a gray scale image so that the gray value at each pixel shows the likelihood of the pixel belonging to the skin. This gray image can be further converted into a binary image showing skin regions and non-skin regions as we describe in the next section.

4.3 SKIN COLOR SEGMENTATION

Starting with the original image in which we want to detect a face, we transform it to a skin-likelihood image (**Figure 4.5**), but before then we have to convert the original image into the chromatic color space (**Figure 4.4**).

This skin-likelihood image is a gray-scale image whose gray values represent the probability of one pixel being a skin region based on the skin color model. This probability is the implementation of the algorithm represented by formula (3.5) on section 3.3 of chapter 3.

Where $x = (r\ g)^T$, m is the mean vector, and C is the covariance matrix.

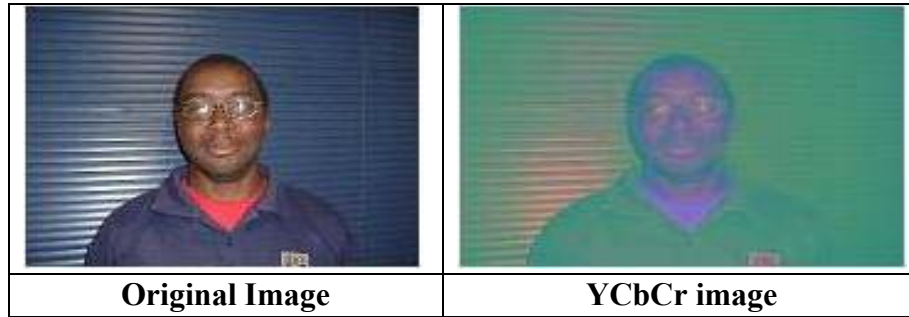


Figure 4.4: shows the RGB image and the equivalent YCbCr image

In **figure 4.5**, all skin regions are shown brighter than the non-skin regions.



Figure 4.5: shows the original image next to the skin-likelihood image

After determining the probability of one pixel belonging to a skin region based on the gray level, further segmentation is done to the gray scale image to threshold the grayscale skin likelihood image into a binary image showing skin regions and non-skin regions. Since skin colors vary between people, an adaptive threshold process to find the optimal threshold value is used. If the threshold value is too low, the amount of segmented skin regions increases. Based on this assumption, an adaptive threshold is created. This method starts the threshold value at 0.65 and decrements by factors of 0.1 until it gets to 0.05. The program determines the optimal threshold value by finding the point when the change in the number of segmented regions is minimum. After the optimal threshold value is determined, the grayscale image is then converted to binary

(figure 4.6). In the binary image (figure 4.6), the white regions show the skin regions and are labeled as 1 and the black regions show the non-skin regions and are labeled as 0. Each of these skin regions is assigned an integer value, and they are tested individually to see if it represents a face.

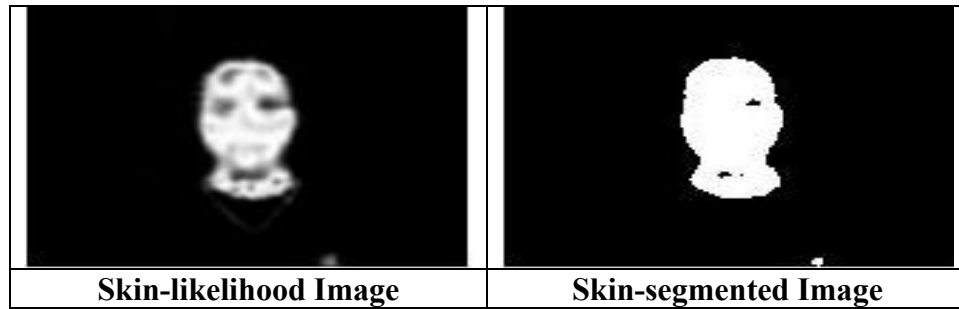


Figure 4.6: Shows the skin-likelihood image and the skin-segmented image

4.4 SELECTION OF FACE REGIONS

We used exactly the same steps as the ones used on **section 3.4 of chapter 3** to determine which regions can possibly determine a human face. Therefore, for further details you can refer to **section 3.4 of chapter 3**, entitled *Selection of Face Regions*.

Another parameter that is also useful to determine if the segmented area is a face is the ratio of height to width. The height to width ratio of a human face is normally 1. In order to improve the detection rate, we found experimentally that 0.8 is a good minimum value. Values below 0.8 are not helpful to conclude that the region is a face, because human faces are oriented vertically. We determined a good upper limit to be 1.6. However, there are situations where we have indeed human faces but the upper limit of the ratio is higher. This happens when the neck of the person is uncovered. Therefore, to account for this we set the ratio to be 1.6 and we eliminate the region below.

4.5 GENERATION OF AVERAGE FACE MODEL

To take the final decision for determining whether the skin region represents a face or not we used the same 2-Dimensional template matching algorithm used on **section 3.5** of **chapter 3**.

Once every region has been successfully evaluated, the original color image is displayed (**Figure 4.7**) with rectangles placed around the faces in the image. This is done by obtaining the coordinates of the part of the image that has the template face (model). With this coordinates, we draw a rectangle in the original color image.

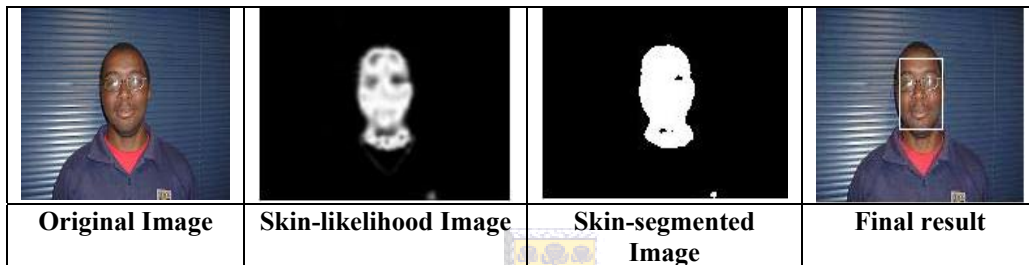


Figure 4.7: General Overview of the described System

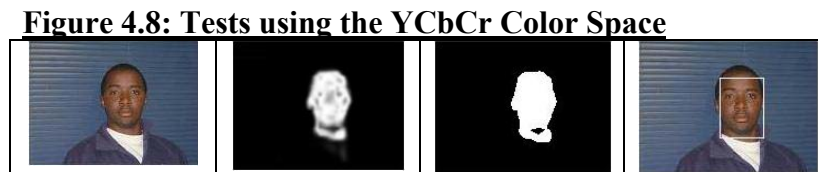
4.6 EXPERIMENTAL RESULTS

The system was tested on a set of 51 images, of who Set 1 are Caucasians, Set 2 are Africans, and Set 3 are Asians, both males and females. This selection of various ethnic groups in the test set was to ensure the robustness of the system. All the images include varying illumination conditions and different background conditions. **Figure 4.8** illustrates the system's face detection capability using the YCbCr color model, from (a) to (d), the first images are the original test images that are being tested for instances of a face, next are the skin likelihood images (the lighter the area in this image, the more likely that that region is a skin region). The third images are the segmented binary images; the last images are the original images with a box drawn around the face. The

program worked relatively well in example 1 of **figure 4.8**, where in the image we have an instance of a face of a dark skin color person. The system successfully recognized the single instance of a face in the image, but it also included the area of the neck. This happened during the labeling of each area, the neck was included with the face area as one labeled region.

Again example 2, where we have an image of a Caucasian lady, shows an instance of a single face, and for this case the system worked very level, it successfully detected the single instance of a face without including the region of the neck.

However, in example 3, which is a demonstration of how multiple instances of faces can be detected, the system failed to detect the second face; instead it falsely detects a bright area on the corner of the image as if it was a face. Reason being that the algorithm labels and examines white regions that have at least one hole in it as probable face candidates. Thus, in example 3, **figure 4.8 (c)**, we see in the segmented white areas that the second face in the background of the image is segmented as a white region with no black holes in it. But, the bright light on the top left corner of the image is segmented and labeled as a white region with two black holes in it. That is why it falsely detects the bright light and misses the second face in the image. Note here also that the segmentation results of the predominant face of the image are not satisfactory. Even though, it successfully recognizes that instance of a face, large parts of the face are labeled as dark a region, because the skin likelihood values are not so bright. Thus, it segments only the edges of the skin areas.



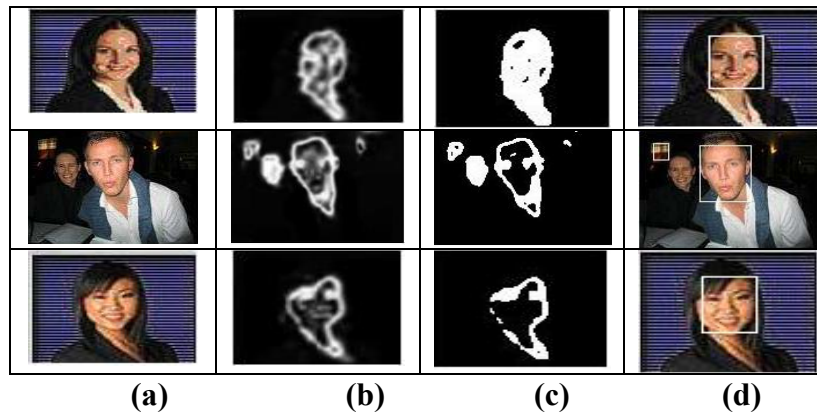


Figure 4.8: (a) The original color images; (b) Skin-likeness images; (c) skin-segmented images; (d) final results

Example 4 of **figure 4.8**, where the image shows a single instance of a face of an Asian (Chinese) lady with her hair covering part of her forehead, the program successfully detects the instance of the face, without including part of the neck in the detection results and also the hair on the forehead does not affect the final result. But the segmentation results of image **figure 4.8 (c)**, shows that the program does not segment the face very well, because it segments only the edges of the skin areas. Creating therefore a big dark region inside the edges of the face.

Table 4.1 gives the face detection results for the YCbCr technique, 57 faces are correctly detected, 9 are missed and 4 faces are falsely detected. The likely reasons for the faces that are missed result from bad lighting conditions and background distractors.

Set of color Images	Correctly detected	Missed	Falsely detected
Set 1	26	1	1
Set 2	19	3	2
Set 3	12	5	1
Total	57	9	4

Table 4.1. Performance of algorithm with YCbCr model

Another reason for the undetected face is due to the very different color of skin or face profiles found across several subjects.

Due to the fact that the program is not completely accurate, we had to run some additional tests to determine the accuracy of this approach. Thus, we found that the optimal values for the height to width ratio are on the range from 0.9 through to 1.9. Meaning that 0.9 is the minimum value and 1.9 is the maximum value of the height to width ratio. The maximum number of positive detections was obtained in this range. In the cases where the ratio was below 0.9 or above 1.9, were usually misses or some faces were detected with either large part of the neck or the hair, it happened also mainly for false detection of other body parts.

The optimal threshold values varied from 0.1 to 0.6, but 0.5 was the optimal threshold value with the highest percentage of successful detection.

During these tests we observed experimentally that the best correlation value is 0.7, because at this value the least amount of non-faces is found and the highest percentage of faces is detected. And the correlation values incremented from 0.2 to 0.8.

The testing was done on 51 images with a total of 66 faces in it with the objective of determining the percentage of faces that are successfully detected with this method. We observed that with the optimal value for the correlation of 0.7 and the ratio of height over the width being on the interval from 0.9 to 1.9, the percentage of faces detected by this approach was 86 % (57 faces). Making this as the best performance. The face is considered detected if it is inside the final result box. In some test examples more than one face was included in the final test result box, and included these faces as being detected. In some images the final result box only contained part of the person's face, for example the chin is cut out but the other facial features are included. In some images, non-skin regions were also detected, because of having color similar to that of the skin.

4.6.1 Computing Time

Our data shows that the runtime for this approach scales linearly with the number of people in the test image (for N people, $runtime \approx (N + 1) \times 400ms$). For real-time performance, a face detection domain is limited to 1 person on an image with face size between 40×40 to 70×70 pixels (around 0.8 seconds) when running the detection code on handheld computers.

The memory footprint of the face detection algorithm is $(3 + N \times 0.006)$ MB for N people. This scales to just over 3MB for 20 people, which is reasonable for handheld devices.

The runtime during the training phase for our 23 skin samples to build the skin color model is around 5 seconds. And we did not need to train non-skin samples, yet the performance is 89 %. And the memory footprint for the training phase of the 23 skin samples to build the skin color model is 3.138MB.



The code size for this method is around 16.723KB.

4.7 DISCUSSION

We observed that the program is more accurate with images with a single face and a solid and homogenous background, even though it works relatively well with images with multiple faces and inhomogeneous background. The method is fast and is reliable, from our experiments it is about 86 % accurate. This program works very well with images where the light incident in it makes it very bright, however its performance drops for the test examples where the face has shade in it. Its segmentation process is also not very efficient in cases of images with the background with colors similar to that of the skin.

In images where two or more instances of faces were very close to each other, during the segmentation period, the program thresholds the face areas and combine them as if it was one big face, thus affecting the detection result as well, and drawing the final result box around two or more faces.

A standard deviation of the pixel gray levels for the face candidates could be used to remove non-faces caused by uniform skin-color-like regions, such as lights, clothes, walls. Non-face templates, such as hand, arms, and neck templates could be implemented to remove other body parts. And additional face templates could be used to detect the missing faces.

4.8 COMPARISON BETWEEN THE HSV APPROACH AND YCbCr TECHNIQUE

For color images, it is proven that it is possible to separate human skin regions from complex background based on either HSV or YCbCr color spaces⁴ [91].

Nevertheless HSV color map is the most adequate for differentiating the skin regions from the contents of the rest of the image [92]. By plotting the skin regions vs. non-skin regions in H vs. S, we determine a set of equations (bounding equations) that maximize the amount of skin pixels and minimize the number of background pixels as illustrated in **figure 4.9**.

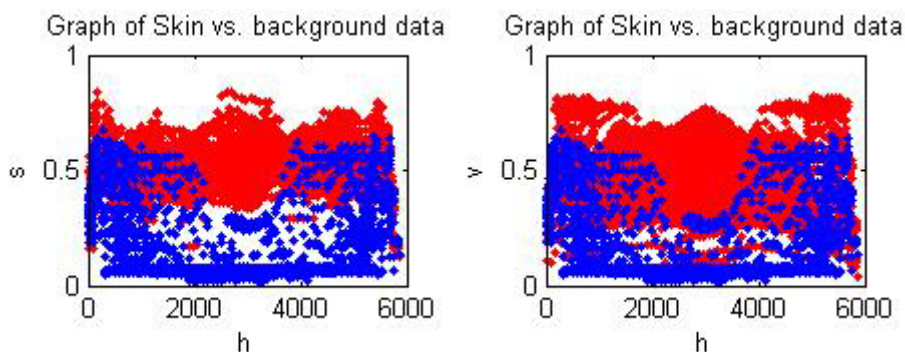


Figure 4.9: Skin data (blue) vs. background data (red) in HS and HV color space.

⁴ We have not evaluated experimentally the other color spaces, besides HSV and YCbCr.

From the above plots of the H-S space and the H-V space of a test image (**figure 4.9**) we observed that there is less overlapping of the non-skin pixels (background) with the skin pixels in HSV space. It is important to note that we implemented our method using the H-S space, we did not consider the H-V of the S-V space, we only illustrate those results because they enabled us to draw a concise conclusion.

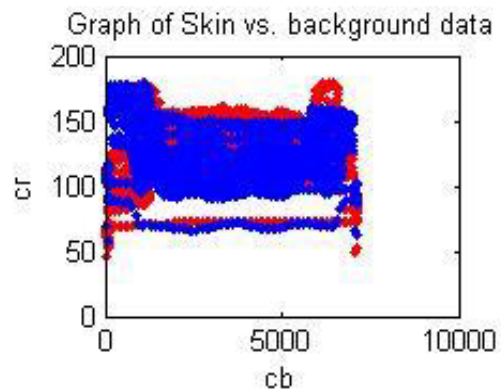


Figure 4.10: Skin data (blue) vs. background data (red) in CbCr color space.



However, we observed from the above plot, of Cb vs. Cr, that there are a lot of background pixels (red) that occupy the same space as the skin pixels (blue) in YCbCr color space.

Nevertheless, we found that by plotting the skin pixels versus the background pixels in the S-V space there are also many background pixels (red) that occupy the same space as the skin pixels (blue) as illustrated in **figure 4.11**.

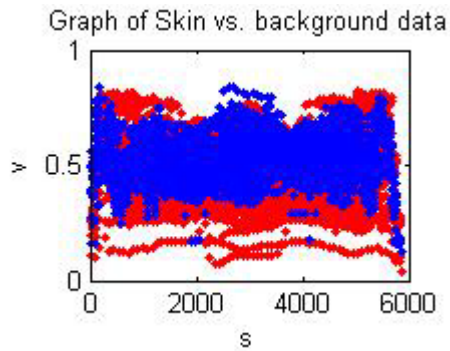


Figure 4.11: Skin data (blue) vs. background data (red) in S-V color space.

Considering that the color of some background contents in images is similar to skin color, and based on the information illustrated in **figure 4.9** and **4.10**, it is very convenient to get rid of the skin-color-like pixels in the HSV color space as compared to YCbCr color space, because those skin-color-like pixels in the HSV color space fall mostly between H values of 0.1 and 0.2, while most skin pixels are less than 0.1 and above 0.8 (**figure 4.12**).

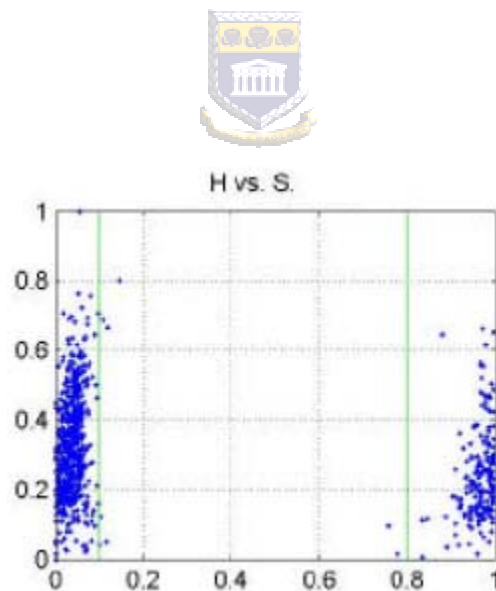


Figure 4.12: Plot of the skin pixel samples and the bounding equations.

Therefore, by setting a threshold at 0.1 we reject the background pixels that are similar to skin pixels in the HSV model, but we cannot do the same with the YCbCr model because the background pixels fall right on the skin area. The above observation is

described experimentally by **figure 4.13**, where we have an image with two instances of faces and a background light that has skin-color-like pixel values.

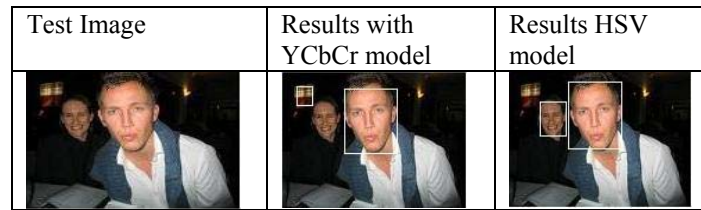
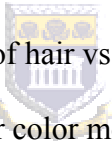


Figure 4.13: Experimental results of how the HSV method handles skin-color-like pixels as compared to the YCbCr method.

From the above **figure 4.13**, we observed that the HSV model approach is more adequate to get rid of the undesirable skin-color-pixel as compared to the results obtained by the YCbCr method, because the HSV correctly detects the two instances of the faces in the image and ignores the rest, but the YCbCr missed one instance of a face and falsely detects the background light that has pixel values similar to those of skin color.

In **figure 4.14**, we illustrate a graph of hair vs. skin samples using Cb and Cr values to demonstrate the advantage of the YCbCr color model in differentiating between the hair and the skin as compared to the HSV color space.



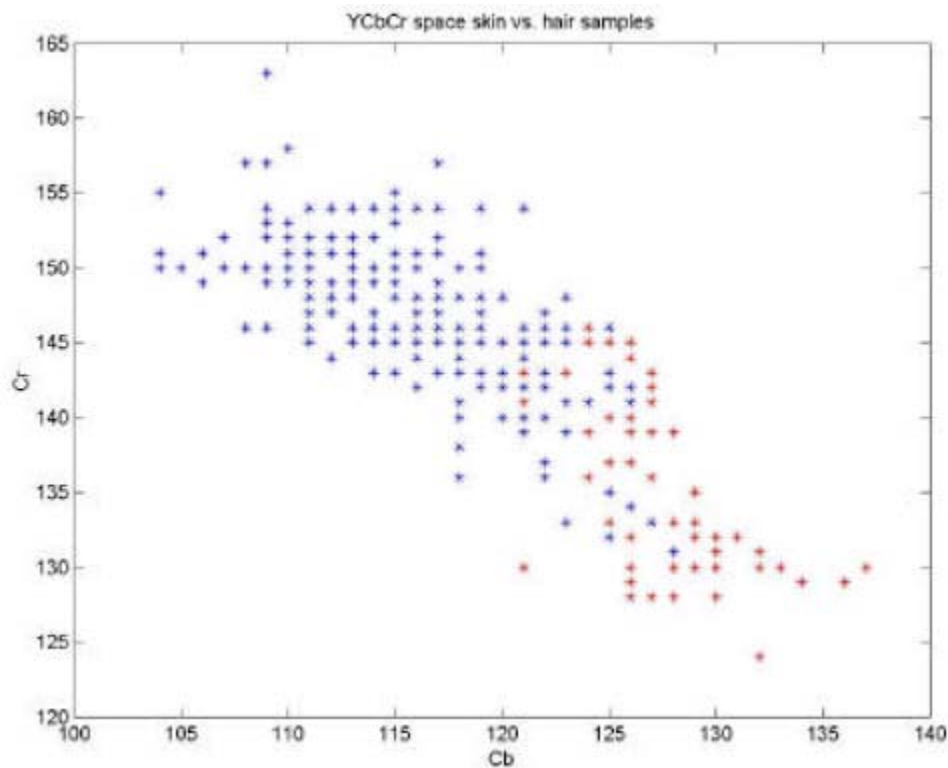


Figure 4.14: Skin (blue) vs. hair (red) pixels.

From the above picture (**figure 4.14**) it shows that the YCbCr color space has a very clear differentiating line between the hairs vs. skin pixels. Thus, a Cb value that gives the best tradeoff of taking out enough hair pixels vs. leaving sufficient skin pixels can be experimentally chosen. Plots of hair vs. skin samples indicated that it is difficult to differentiate them in the HSV color space.

To calculate the percentage of faces detected by both methods, we ran tests on 51 distinct images with 66 faces in the images. The percent of faces included in the final result box for the method using HSV skin color model was 89 % (59 faces), and for the one using YCbCr skin color model was 86 % (57 faces).

We observed also that our segmentation technique has some limitations. One stems from the fact that facial features (mouth and eyes) fall in the same color space area as the skin color both in HSV and YCbCr. Therefore, it is difficult to make these facial features

more prominent in order to increase the correlation with the face template model.

Another limitation is observed with some hair colors like blonde or light brown.

4.9 SUMMARY

In chapter 4, we have described in detail another low complexity algorithm for face detection using skin color modeling in the YCbCr color space, we then presented the results of our experiments that has enabled us to deduce a number of conclusions and make a comparison between this approach and the one described in chapter 3.



CHAPTER 5

FACE TRACKING

5.1 INTRODUCTION

In this chapter the focus is on tracking the face in 2D color video streams using HSV (Hue Saturation Value) as the histogram color space. HSV was chosen because it offers the advantage which is the separation of chroma information from intensity information. The same could have been done using YCbCr as the histogram color space. The program is used to compute 3D trajectories for a remote surveillance system. This is an implementation of the paper by [93].

In a complex background, a robust, and reliable visual tracking of an object will require the combination of several different visual modules, each using a different criterion and each employing different assumptions about the incoming images. These modules are chosen as much as possible to be orthogonal to each other so that when one module fails the other one can come to its aid.

From mathematical elementary set theory, every closed set in the plane can be decomposed into two disjoint sets; the boundary and the interior [94]. Considering that these two sets are complementary, one can reason that the failure modes of a tracking module focusing on the object's boundary will be orthogonal to those of a module focusing on the object's interior.

A method for face tracking that integrates the output of two modules is presented: one that matches the intensity gradients along the face's boundary and one that matches the color histogram of the face's interior in the HSV color space.

Even though the gradient and color module are complementary, they operate in a symmetric fashion, thus, making the integration step trivial.

The result is a robust and reliable face tracker that is accurate and insensitive to out-of-plane rotation, tilting, severe but brief occlusion, arbitrary camera movement, and multiple moving people in the background.

5.2 FACE TRACKING METHOD

Assuming that the head can be modeled as an ellipse; therefore the head's projection onto the image plane is modeled as an ellipse whose position and size are constantly updated by a local search combining the output of a module concentrating on the intensity gradient around the ellipse's perimeter with that of another module focusing on the color histogram of the ellipse' interior. Thus, the tracking system needs only the trajectory of the head's center of mass.

The face detection method presented on chapter 3 is used to estimate the position (x, y) and the size σ of the face in an image. The face is modeled as a vertical ellipse with a fixed aspect ratio of 1.2, such that (x, y) is the center of the ellipse and σ is the length of the minor axis.

A two module function was proposed and is evaluated in a given window to find the best ellipse that contains the head. One of the modules deals with color in the ellipse and the other module deals with the intensity gradient across the ellipse boundary. We have therefore to maximize a function inside the window to locate the optimum. The search space is made of three parameters, namely x , y and σ . X and Y correspond to pixel coordinate, and σ measures the width of minor axis of the ellipse. The ratio between the major and minor axes in the ellipse is set at 1.2 and it is based on the following equation:

$$s' = \arg \max \{ \overline{\phi}_g(s_i) + \overline{\phi}_c(s_i) \} \quad (5.1)$$

The search space or window (S) that gives the best results was found to be 4x4x1, that is; +/-4 pixels in the x direction, +/-4 pixels in the y direction and +/-1 pixel wide.

The gradient and color module are defined respectively by the following expressions:

$$\phi_g(s) = \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} |n_\theta(i) * g_s(i)| \quad (5.2)$$

$$\phi_c(s) = \frac{\sum_{i=1}^N \min(I_s(i), M(i))}{\sum_{i=1}^N I_s(i)} \quad (5.3)$$



These equations can be normalized by subtracting the minimum value found in the window and dividing by the range described by equations (5.4) and (5.5):

$$\phi_g(s) = \frac{\overline{\phi_g}(s) - \min_{s_i \in S} \phi_g(s_i)}{\max_{s_i \in S} \phi_g(s_i) - \min_{s_i \in S} \phi_g(s_i)} \quad (5.4)$$

$$\phi_c(s) = \frac{\overline{\phi_c}(s) - \min_{s_i \in S} \phi_c(s_i)}{\max_{s_i \in S} \phi_c(s_i) - \min_{s_i \in S} \phi_c(s_i)} \quad (5.5)$$

The gradient module sums the dot product of the normal vector with its gradient (using Sobel edge detector) along all contour pixels.

The color module compares the histogram at ellipse location s with a model histogram. This histogram intersection is calculated by summing the value of the lesser bin between the model histogram and the image histogram for all the bins in the histogram. It is important to stress that this only works if the histogram are normalized; otherwise the module will be biased towards ellipses that are smaller than the one used to obtain the model histogram.

The modified color module equation is then expressed by:

$$\phi_g(s) = \sum_{i=1}^N \min(\bar{I}_s(i), \bar{M}(i)) \quad 5.6$$

$$\bar{I}_s(i) = \frac{I_s(i)}{\sum_{i=1}^N I_s(i)} \quad 5.7$$



The modified model histogram is done by the following expression:

$$\bar{M}_s(i) = \frac{M_s(i)}{\sum_{i=1}^N M_s(i)} \quad 5.8$$

The color histogram using the HSV color space is well suitable for the tracking task because of its ability to implicitly capture complex, multimodal patterns of color. Moreover, because it disregards all geometric information, it remains invariant to many complicated, non-rigid motions.

Off-line, the subject presents three-quarters view of the camera in order to capture the face, and a model histogram is built by counting the pixels inside the ellipse (the ellipse can be manually placed or automatically placed via the gradient module). At run time, the histogram intersection [40] is computed between the model histogram and the image histogram at each hypothesized location.

Our color space consists of scaled versions of the three axes $H - S$, $S - V$ and $H + S + V$. the first two contain the chrominance information and are sampled into eight bins each, whereas the last one contain the luminance information and is sampled into four bins.

On our face detection method presented and examined previously we have discarded the luminance component completely due to the fact that we used skin color as the basic feature for face detection and the luminance component is not homogenous across the person's face because of ambient lighting. However, for the tracking system we have utilized a little bit of the luminance information for out-of-plane rotation, because, based on chrominance alone, dark brown hair looks similar to a white wall.

5.3 EXPERIMENTAL RESULTS

This algorithm was tested on four video sequences. The format of the sequences is .avi, and the original resolution is 240 x 320. We established the frame rate to be 25 frames per second and sizes of the faces were around 40 x 30. In all video sequences the camera is not static and it includes translation, and rotation.



Figure 5.1: Some frames of the image sequence shown below

In the uncluttered environment shown in **figure 5.1**, we observed that the gradient module was able to consistently track the subject's slowly-moving face for about 300 frames without being influenced by the background. However, in cluttered environments the gradient module fails too often because the ellipse tends to become attracted to gradients inside the face.

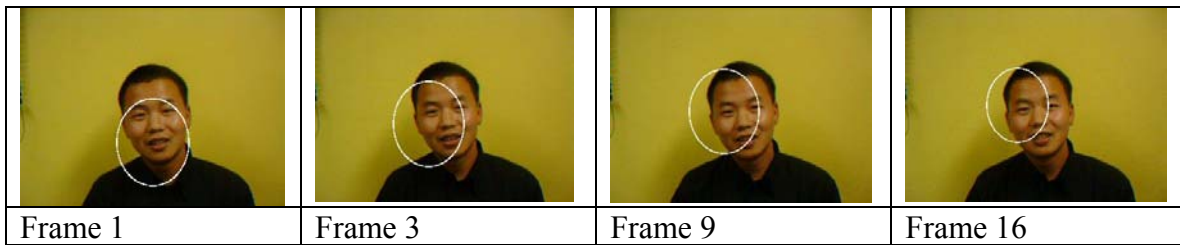


Figure 5.2: Some frames of the image sequence shown below

In **figure 5.2** we have an image sequence of an Asian male in an uncluttered environment again, with a skin-colored background. The subject is able to move without causing the tracker to become lost. Even though the ellipse location is unstable, yet it remains in the subject face throughout the tracking process because of the non-skin pixels such as the hair, eyes, and mouth because the color module tries to compensate for this based on the results from the model histogram. If the subject subsequently moves away from that area to an area with no skin colored background, the ellipse's location quickly stabilizes onto the face. The color module was found to sometimes slip down to the subject's neck, a problem that is solved by adding the gradient module, which tends to get a strong response from the outline of the top of the face.

The color module is very important to handle background clutter, because it has the advantage of having a large basin of attraction. For example we noticed a case in which the ellipse was barely hanging on to the head because of the subject's quick acceleration

and the skin-colored area behind the subject. Although the gradient module was distracted by the background and tried to pull the ellipse to the left, the color module correctly pulled to the right, even though a large percentage of the ellipse's interior contained the distracting skin-colored background.

With multiple people in the scene the tracker usually succeeded but was distracted when the two faces occupied adjacent regions in the image. There was a scenario where the ellipse temporarily preferred one of the faces in the background but quickly returned to the subject when his continued motion caused the other person to be occluded. Had the subject changed the direction at that point in time, we believe that the tracker would have lost him.

5.4 DISCUSSION

We observed experimentally that the face tracker is robust and accurate and produces satisfactory results, even though in few occasions it caused it to lose the subject's face and then it failed to recover from it. We also noticed that the gradient module seems to get easily distracted with the background making the ellipse not fit the face as strongly as it is supposed to be. Robustness is attained by using two orthogonal modules, one based on the intensity gradient around the face's perimeter and the other based on the color histogram of the face's interior.

The color module greatly helps the gradient module by ignoring background clutter, correctly handling changes in scale, and providing a larger basin of attraction. In a similar way, often the gradient module helps the color module.

5.5 COMPARISON WITH OTHER FACE TRACKING APPROACHES

This face tracking method can handle significant out-of-plane rotation, textured foregrounds and backgrounds, and multiple moving people in the background, all simultaneously.

Neural network-based or template-based trackers cannot handle severe out-of-plane rotation because such a rotation causes the face to disappear. The color-based approaches also tend to have problem with skin-colored objects or other people in the background.

Face tracking techniques using some form of background differencing either require a static camera or restrict the camera's motion to rotation about its focal point. Many of these approaches perform motion-based figure-ground segmentation that tends to fail when the camera zooms or when multiple objects move in the image. Nevertheless, reliable tracking can be performed by combining a template-based face tracking method with stereo depth. But, besides the additional hardware required, it is not clear whether this system would be able to handle multiple people at a similar depth as the subject.

Some of the methods cited above contain multiple modules, nonetheless our methods is set apart from them because it can handle out-of-plane rotation and a dynamic camera simultaneously.



5.6 CONCLUSION

We have presented a method for human face tracking in video sequences, which is exclusively based on two orthogonal modules, the gradient module and the color module. Since these modules have orthogonal failure modules, they serve to complement each other. The modules are used to compute 3D trajectories for a video surveillance application. Tests on these sequences indicated a tracking success rate of around 94% was achieved on image sequences of about 300 frames at a frame rate of 25 frames per second.

A Gaussian filter could be used on the image before calculating the gradients and it would be helpful to improve the results of the face tracker. We also recommend the use of a dynamic window size to help deal with fast movement of the face. The optimal value

could be computed from the velocity of the face using results from the previous frames. We also recommend the implementation of an additional adaptive histogram module, because the current color module is not adaptive and it causes the color module to become confused when there is changing in the lighting conditions or when there is automatic gain adjustments by the camera.



CHAPTER 6

CONCLUSION AND DIRECTIONS FOR FURTHER RESEARCH

6.1 INTRODUCTION

The main focus of this thesis has been to investigate low complexity face detection and tracking algorithms for use in remote surveillance with mobile phones and other handheld devices. To our knowledge mobile phones have not been commercially used in the very important task of remote monitoring/surveillance. The most probable reason is the issue of computing capability of handheld devices, as well as memory space and battery charge.

Chapter 3 describes in detail our proposed solution, a low complexity face detection algorithm that uses skin color as the basic feature for the detection process, we then presented the results of our experiments and drawn a number of conclusions. Furthermore in chapter 5 we present our proposed solution for the face tracking problem in which the face's projection onto the image plane is modeled as an ellipse whose position and size are consistently updated with a local search combining the output of an intensity gradient module and the color histogram module.

The aim of this chapter is to provide answers to the research questions and to provide some conclusive remarks of the findings of the study, as well as highlight some areas of application for this method. Furthermore, we discuss directions for further research.

6.2 RESEARCH QUESTIONS

The research questions that guided our research and the corresponding findings are described in the same order as outlined in section 1.4.

1) Examine the advantages and disadvantages of having the face tracking system either on the hand held or on a server to which the device has access to it?

On a remote server, a face could be tracked quite easily, but it has to be tracking many faces from many hand held devices at the same time and the hand held device must send to it a face or frame containing the face. The server must then inform the hand held device if the face is that of the owner of the asset or an intruder. It is a complex operation but a client/server architecture using TCP sockets is suitable for this approach. Where a threaded server spawns off processes to handle all the requests from the clients simultaneously. The number of spawned processes is the same as the number of requests from the clients. For this approach, there are no problems with computing time, disk space, battery time or memory.

By modern standards, the operational portion of face tracking systems with low complexity algorithms is fairly trivial, thus, it is not a problem to put it on a hand-held device. To build the skin color model using the HSV color space it takes around 5 seconds of runtime of the 23 skin samples, and the memory footprint scales to up to 3.138MB for the 23 skin samples. Thus, the training phase using skin color does well on a handheld. But for more complexes like the Convolutional Neural Network architecture described in chapter 5, the training phase would not do well on a handheld. However, it can be trained offline, and their on-line application could be computationally easy with further preprocessing.

It matters a lot how many faces you want to compare and what sort of error rate you are comfortable with. There are also some concerns with getting high quality images, and memory capacity to store the gallery of images for comparison.

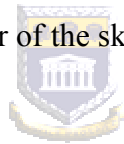
Store streaming video to hard drive, requires large amount of disk space, it might be a disadvantage for a handheld device as compared to a server that the device has access to it.

2) Does the colour of the face affect the tracking results?

The tracking results are not affected by the color of the face, and this stems from the fact that in both cases, for the HSV model and YCbCr, we created skin color models and trained it with skin samples of individuals of all races, namely Africans, Asians, and Caucasians. Therefore, in any test image where appears a color similar to that of the skin the program automatically segments it, then passes it over to be evaluated for other characteristics to determine if it is indeed an instance of a face or not.

We observed experimentally as well that in every circumstance where there was a face, irrespective of the skin color it was always segmented and labeled as a skin region, it might have been missed due to other aspects such as the height to width ratio or any other factor but not because of the color of the face.

For the tracking, because we used the gradient and the color module and the fact that they complement each other, makes it independent of the color of the skin, meaning the results were always the same, the color of the skin was not a factor.



3) Does the time of tracking influences the tracking results?

The color distribution of skin-color of different people is clustered in a small region of either the HS or the chromatic color spaces. This has been proven experimentally for color images. Thus, the process of separating human skin regions from complex backgrounds is not cumbersome. However the time of tracking does not affect the tracking results, because the segmentation technique applied uses skin color as the basic feature to separate skin regions from non-skin regions in a color image and even though the color histogram model used is not adaptive it can cause the color module to become confused when there is quick changes in lighting condition, it is still not a major factor that could affect the tracking result, specially if the lighting conditions are stable, and there is no sudden changes.

This limitation can be overcome by implementing an additional adaptive histogram module that is able to quickly update itself due to sudden changes in lighting conditions.

4) Can we network the surveillance camera?

The simultaneous development of Broadband Wireless Access (BWA) network equipments, video surveillance technologies, and other technological improvements has brought wireless IP cameras or wireless network cameras into the surveillance world.

With an IP surveillance camera such as **D-Link Securicam Network™ DCS-5300W Internet Camera**, you can connect to a wired Ethernet or Enhanced 802.11b wireless network to provide high quality video and audio. Each camera supports networking protocols such as TCP/IP, HTTP, SMTP, FTP, Telnet, NTP, DNS and DHCP. It provides advanced technology with DDNS and UpnP support.

From the Web browser you can view video from the camera and you can manage the camera. The camera has an auto pan mode that enables it to move 270 degrees horizontally and the patrol mode cycles through up to 20 preset positions.



Its built-in software enables it to act as a Web server, it also further enhances the security of the camera, allowing you to archive streaming video to your hard drive, search and playback stored video, monitor as many as 16 cameras in a single screen, and set up motion detection to trigger automatic recording with e-mail alert notification.

It uses the latest enhanced 802.11b wireless technology to communicate wirelessly with a maximum wireless signal rate of up to 22Mbps⁵ and 256-bit WEP security encryption.

Therefore, and based on the facts prior stated, the surveillance cameras can be networked. But an efficient handshake protocol to network the cameras is required. A

⁵ Maximum wireless signal rate derived from IEEE Standard 802.11b specifications. Actual data throughput will vary. Network conditions and environmental factors lower actual data throughput rate.

camera handshake protocol [95] is an algorithm that can track objects across video frames from many networked cameras in a way that when the being object tracked appears in the next camera, it is handed over such that tracking carries on.

5) What triggers the surveillance cameras?

Motion detection is used to trigger the surveillance cameras.

Advanced digital video motion detection systems have 3-D capabilities, therefore they are able to determine the size and direction of an object, to identify and screen out normal repetitive actions such as the movement of fountains by analyzing the recorded tracking history, and compensate for camera vibration.

A practical example would be that motion detection systems trigger alarms when people travel in the wrong direction at certain times of the day, such as the owner of a car could set up the system to trigger the alarm if it detected someone approaching the car when he is not in the car. Nevertheless, special attention needs to be paid to false alarms.

Moreover, Cameras like **D-Link Securicam Network™ DCS-5300W Internet Camera** contain built-in motion detection, which require an external sensor or motion detection that could be used to trigger the surveillance cameras.

6) How should we initiate automatic triggering to avoid false alarm?

A false alarm is defined as an alarm condition that results from something other than a break-in, or an actual emergency situation.

Since we use motion detection to trigger the surveillance cameras, one type of false alarm is that caused by the motion of large objects. To avoid it, the video frame is split into four quarters and the human detection algorithm is used in each quarter separately.

And a threshold is set, such that if three or more of the frame quarters have human motion characteristics, then human motion is declared.

Another solution would be to use equipment that supports the sensitivity adjustment and the noise signal tolerance to avoid false alarm.

6.3 OBSERVATIONS

We observed experimentally that the color distribution of skin color of people of all ethnicities is clustered in a small region of both the HSV color space and the YCbCr color space for color images. It is possible to separate human skin regions from complex background using either HSV or YCbCr color space. Moreover, the HSV color map is the most adequate for differentiating the skin regions from the contents of the rest of the image, because there is less overlapping of the non-skin pixels with the skin pixels, if you make a plot of the skin pixels vs. the background in the HSV space. The HSV color space is also more adequate to get rid of the undesirable skin-color-like pixels of the background. However, the YCbCr method better differentiates between the hair and the skin as compared to the HSV color space.

The percentage of faces detected by our approach using the skin color model in the HSV color space was around 89 %, using the best value for the correlation of 0.6 and the height to width ratio being the range 0.8 to 1.9. For the YCbCr skin color model the percentage was around 86 %, with the optimal value for the correlation being 0.7 and the height to width ration on the range between 0.9 and 1.9.

The memory footprint scales to just over 3MB for 20 people during the operational portion and 3.138 MB to build the skin color model of 23 skin samples, which is reasonable for handheld devices.

The runtime to build the skin color model is around 5 seconds, and the real-time performance when running the detection code on a handheld is around 0.4 seconds per image.

We have proposed and implemented a fast and reliable algorithm for face detection based on the HSV-colour space, with adaptive threshold which simplify the detection of faces within video sequences on mobile phones and that are suitable for implementation on mobile handsets. The algorithm is used to segment and detect multiple faces in images.

In terms of clarity of segmentation, by mere observation we found the HSV to perform much better as compared to YCbCr. The HSV method is also superior in accuracy in test images of people of lighter completion. The HSV also performs better in images with shade in it. However, in very bright images the YCbCr approach does well as compared to the HSV model.



We found that the conversion from RGB space to HSV is more complex than the conversion from RGB to YCbCr.

Background distractors affect both methods, and other frequent misses included regions with very similar skin likelihood values.

Furthermore, we observed experimentally that the face tracker is robust and accurate and produces satisfactory results, even though in few occasions it caused it to lose the subject's face and then it failed to recover from it. We also noticed that the gradient module seems to get easily distracted with the background making the ellipse not fit the face as strongly as it is supposed to be. Robustness is attained by using two orthogonal modules, one based on the intensity gradient around the face's perimeter and the other based on the color histogram of the face's interior.

The color module greatly helps the gradient module by ignoring background clutter, correctly handling changes in scale, and providing a larger basin of attraction. In a similar way, often the gradient module helps the color module.

This application is target for mobile handsets such as the NOKIA 9500 Communicator. The Nokia emulator can be integrated with Sun One Studio and Nokia Developers Suite. The emulator allows us to test the application as if it is being tested on an actual mobile handset.

6.4 APPLICATIONS

This face detection and tracking technique, besides being used as a prior step for face recognition using handheld devices in remote surveillance, it could also be used in a wide range of applications such as a human motion detection system to avoid false alarms in secure areas, gesture recognition to bridge the digital divide that exists with the deaf community, security control, detection and tracking of online pornographic images[96], video retrieving, image database management, biometric signal processing, human computer interface, face recognition.

6.5 DIRECTIONS FOR FURTHER RESEARCH

There are a number of directions for future work. Our proposed method is not completely accurate, we obtained a detection rate of 89 %, and therefore, further study should be done to improve the detection rate. The combination of HSV and YCbCr methods should be investigated to eliminate the detection errors originated by the inadequacy of the HSV model to differentiate very clearly between the hair and the skin. A combination of the two methods would also be helpful to improve the detection of images very bright or with shade in it.

Non-face templates, such as hand, arms, and neck templates could be investigated and implemented to remove other body parts. And additional face templates could be

used to detect the missing faces. The use of a standard deviation of the pixel gray levels for the face candidates should also be investigated in order to remove non-faces caused by uniform skin-color-like regions such as light clothes, walls, etc.

A Gaussian filter could be used on the image before calculating the gradients and it would be helpful to improve the results of the face tracker. We also recommend the use of a dynamic window size to help deal with fast movement of the face. The optimal value could be computed from the velocity of the face using results from the previous frames. We also recommend the implementation of an additional adaptive histogram module, because the current color module is not adaptive and it causes the color module to become confused when there is changing in the lighting conditions or when there is automatic gain adjustments by the camera.

Porting the source code from MATLAB into a low level programming language such as C or even C++, would improve the performance of this method at runtime, in order for it to be closer to the 25 frames per seconds, which is the threshold to realize real-time computing.

Different solutions such as a Convolutional Neural Network architecture [89] combined with optical preprocessing could be researched to obtain very fast detection/tracking rate. Principal Component Analysis (PCA), Support Vector Machines could also be investigated and tested to correct some of the performance problem that our method contains.



REFERENCES

- [1] Ilias Sachpazidis, “@Home: A System for real-time monitoring of patients' vital parameters,” Proceedings 2002 (2002) Korea-Germany, Joint workshop on Advanced Medical Image Processing 6.2002 Heidelberg, Germany.
- [2] Ming-Hsuan Yang, David J. Kriegman and Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, January 2002.
- [3] Kah Kay Sung and Tomaso Poggio, “Example-based learning for view-based human face detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998), no. 1, 39-51.

- [4] K. Lam and H. Yan, "Fast Algorithm for Locating Head Boundaries", *J. Electronic Imaging*, vol. 3, no. 4, pp. 351-359, 1994.
- [5] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696-710, July 1997.
- [6] I. Craw, D. Tock, and A. Bennett, "Finding Face Features", *Proc. Second European Conf. Computer Vision*, pp. 92-96, 1992.
- [7] H. P. Graf, T. Chen, E. Petajan and E. Cosatto, "Locating Faces and Facial Parts", *Proc. First Int'l Workshop Automatic Face and Gesture Recognition*, pp. 41-46, 1995.
- [8] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey", *Proc. IEEE*, vol. 83, no. 5, pp. 705-740, 1995.
- [9] A. Samal and P. A. Iyengar, "Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey", *Pattern Recognition*, vol. 25, no. 1, pp. 65-77, 1992.
- [10] M. Turk and A. Pentland, "Eigenfaces for Recognition", *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
- [11] C. Kotropoulos, A. Tefa, and I. Pitas, "Frontal Face Authentication Using Variants of Dynamic Link Matching Based on Mathematical Morphology", *Proc. IEEE Int'l Conf. Image Processing*, pp. 122-126, 1998.
- [12] C. Kotropoulos, A. Tefa, and I. Pitas, "Varyants of Dynamic Link Architecture Based on Mathematical Morphology for Frontal Face Authetication", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 814-819, 1998.
- [13] J. L. Crowley and F. Berard, "Multi-Modal Tracking of Faces for Video Communications", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 640-645, 1997.

- [14] T. Darrel, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection", *Int'l J. Computer Vision*, vol. 37, no. 2, pp. 175-185, 2000.
- [15] G. J. Edwards, C. J. Taylor, and T. Cootes, "Learning to Identify and Track Faces in Images Sequences." *Proc. Sixth IEEE Int'l Conf. Computer Vision*, pp. 317-322, 1998.
- [16] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying Facial Actions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974-989, October 2000.
- [17] L. A. Essa and A. Pentland, "Facial Expression Recognition Using a Dynamic Model and Motion Energy," *Proc. Fifth IEEE Int'l Conf. Computer Vision*, pp. 360-367, 1995.
- [18] G. Yang and T. S. Huang, "Human Face Detection in Complex Background," *Pattern Recognition*, vol. 27, no.1, pp. 53-63, 1994.
- [19] C. Kotropoulos and I. Pitas, "Rule-Based Face Detection in Frontal Views," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2537-2540, 1997.
- [20] T. Kanade, "Picture Processing by Computer Complex and Recognition of Human Faces," PhD thesis, Kyoto Univ., 1973.
- [21] H.P. Graf, T. Chen, E. Petajan, and E. Cosatto, "Locating Faces and Facial Parts," *Proc. First Int'l Workshop Automatic Face and Gesture Recognition*, pp. 41-46, 1995.
- [22] H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan, "Multimodal System for Locating Heads and Faces," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pp. 88-93, 1996.
- [23] J. Yang and A. Waibel, "A Real-Time Face Tracker," *Proc. Third Workshop Applications of Computer Vision*, pp. 142-147, 1996.

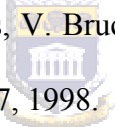
- [24] T. S. Jebara and A. Pentland, "Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 144-150, 1997.
- [25] T. S. Jebara, K. Russell, and A. Pentland, "Mixtures of Eigenfeatures for Real-Time Structure from Texture," Proc. Sixth IEEE Int'l Conf. Computer Vision, pp. 128-135, 1998.
- [26] S. Satoh, Y. Nakamura, and T Kanade, "Name-it: Naming and Detecting Faces in News Videos," IEEE Multimedia, vol. 6, no. 1, pp. 22-35, 1999.
- [27] Y. Miyake, H. Saitoh, H. Yaguchi, and N. Tsukada, "Facial Pattern Detection and Color Correction from Television Picture for Newspaper Printing," J. Imaging Technology, vol. 16, no. 5, pp. 165-169, 1990.
- [28] D. Saxe and R. Foulds, "Towards Robust Skin Identification in Video Images," Proc. Second Int'l Conf. Automatic Face and Gesture Recognition, pp. 379-384, 1996.
- [29] R. Kjeldsen and J. Kender, "Finding Skin in Color Images," Proc. Second Int'l Conf. Automatic Face and Gesture Recognition, pp. 312-317, 1996.
- [30] K. Sobottka and I. Pitas, "Face Localization and Feature Extraction Based on Shape and Color Information," Proc. IEEE Int'l Conf. Image Processing, pp. 483-486, 1996.
- [31] H. Wang and S.-F. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video," IEEE Trans. Circuits and Systems for Video Technology, vol. 7, no. 4, pp. 615-628, 1997.
- [32] D. Chai and K. N. Ngan, "Locating Facial Region of a Head-and-Shoulders Color Image," Proc. Third Int'l Conf. Automatic Face and Gesture Recognition, pp. 124-129, 1998.

- [33] Y. Dai and Y. Nakano, "Extraction for Facial Images from Complex Background Using Color Information and SGLD Matrices," Proc. First Int'l Workshop Automatic Face and Gesture Recognition, pp. 238-242, 1995.
- [34] Y. Dai and Y. Nakano, "Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scene," Pattern Recognition, vol. 29, no. 6, pp. 1007-1017, 1996.
- [35] E. Saber and A.M. Tekalp, "Frontal-View Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Functions," Pattern Recognition Letters, vol. 17, no. 8, pp. 669-680, 1998.
- [36] Q. Chen, H. Wu, and M. Yachida, "Face Detection by Fuzzy Matching," Proc. Fifth IEEE Int'l Conf. Computer Vision, pp. 591-596, 1995.
- [37] M.-H. Yang and N. Ahuja, "Detecting Human Faces in Color Images," Proc. IEEE Int'l Conf. Image Processing, vol. 1, pp. 127-130, 1998.
- [38] J.L. Crowley and J.M. Bedrune, "Integration and Control of Reactive Visual Processes," Proc. Third European Conf. Computer Vision, vol. 2, pp. 47-58, 1994.
- [39] J.L. Crowley and F. Berard, "Multi-Modal Tracking of Faces for Video Communications," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 640-645, 1997.
- [40] M.J. Swain and D.H. Ballard, "Color Indexing," Int'l J. Computer Vision, vol. 7, no.1, pp. 11-32, 1991.
- [41] J. Cai, A. Goshtasby, and C. Yu, "Detecting Human Faces in Color Images," Proc. 1998 Int'l Workshop Multi-Media Database Management Systems, pp. 124-131, 1998.
- [42] S.-H. Kim, N.-K. Kim, S.C. Ahn, and H.-G. Kim, "Object Oriented Face Detection Using Range and Color Information," Proc. Third Int'l Conf. Automatic Face and Gesture Recognition, pp. 76-81, 1998.

- [43] M.-H. Yang and N. Ahuja, "Gaussian Mixture Model for Human Skin Color and Its Application in Image and Video Databases," Proc. SPIE: Storage and Retrieval for Image and Video Databases VII, vol. 3656, pp. 458-466, 1999.
- [44] S. McKenna, Y. Raja, and S. Gong, "Tracking Colour Objects Using Adaptive Mixture Models," Image and Vision Computing, vol. 17, nos. 3 / 4, pp. 223-229, 1998.
- [45] T. Sakai, M. Nagao, and S. Fujibayashi, "Line Extraction and Pattern Detection in a Photograph," Pattern Recognition, vol. 1, pp. 233-248, 1969.
- [46] A. Yuille, P. Hallinan, and D. Cohen, "Feature Extraction from Faces Using Deformable Templates," Int'l J. Computer Vision, vol. 8, no. 2, pp. 99-111, 1992.
- [47] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, 1998.
- [48] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 130-136, 1997.
- [49] V. Bruce, P.J.B. Hancock, and A.M. Burton, "Human face perception and identification," In Face Recognition: From Theory to Applications, H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang, Eds. Springer-Verlag, Berlin, Germany, 51-72, 1998.
- [50] B. Knight, and A. Johnston, "The role of movement in face recognition," Vis. Cog., vol. 4, 265-274, 1997.
- [51] A. Shio, and J. Sklansky, "Segmentation of people in motion," In Proceedings, IEEE Workshop on Visual Motion, 325-332, 1991.
- [52] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland, "Multimodal person recognition using unconstrained audio and video," In Proceedings, International Conference on Audio- and Video-Based Person Authentication, 176-181, 1999.

- [53] S. McKenna, and S. Gong, "Recognising moving faces," In Face Recognition: From Theory to Applications, H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang, Eds. Springer-Verlag, Berlin, Germany, 578-588, 1998.
- [54] L. Gu, S.Z. Li, and H.J. Zhang, "Learning probabilistic distribution model for multiview face detection," In Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2001.
- [55] Y. Li, S. Gong, and H. Liddell, "Modelling face dynamics across view and over time," In Proceedings, International Conference on Computer Vision, 2001b.
- [56] B. Li, and R. Chellappa, "Face verification through tracking facial features," Journal Opt. Soc. Am., vol. 18, 2001.
- [57] Y. Li, S. Gong, and H. Liddell, "Constructing facial identity surfaces in a nonlinear discriminating space," In Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2001a.
- [58] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, 602-604, 1993.
- [59] D. Terzopoulos, and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, 569-579, 1993.
- [60] A. Yuille, and P. Hallinan, "Deformable templates," In Active Vision, A. Black, and A. Yuille, Eds., Cambridge, MA, 21-38, 1992.
- [61] M. Black, and Y. Yacoob, "Tracking and recognizing facial expressions in image sequences using local parameterized models of image motion," Technical report, CS-TR-3401, Center for Automation Research, University of Maryland, College Park, MD, 1995.

- [62] T. Maurer, and C. von der Malsburg, "Tracking and learning graphs and pose on image sequences of faces," In Proceedings, International Conference on Automatic Face and Gesture Recognition, 176-181, 1996b.
- [63] J. Strom, T. Jebara, S. Basu, and A. Pentland, "Real time tracking and modeling of faces: An EKF-based analysis by synthesis approach," Technical report, TR-506, MIT Media Lab, Massachusetts, Institute of Technology, Cambridge, MA, 1999.
- [64] G.D. Hager, and P.N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, 1-15, 1998.
- [65] M. Brand, and R. Bhotika, "Flexible flow for 3D nonrigid tracking and shape recovery," In Proceedings, IEEE Conference on Computer vision and Pattern Recognition, 2001.
- [66] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," In Vision Algorithms: Theory and Practice, Springer-Verlag, Berlin, Germany, 2000.
- [67] D. DeCarlo, and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," International Journal of Computer Vision, vol. 38, 99-127, 2000.
- [68] H. Wechsler, V. Kakkad, J. Huang, S. Gutta, and V. Chen, "Automatic video-based person authentication using the RBF network," In Proceedings, International Conference on Audio- and Video-Based Person Authentication, 85-92, 1997.
- [69] S.J. McKenna, and S. Gong, "Non-intrusive person authentication for access control by visual tracking and face recognition," In Proceedings, International Conference on Audio- and Video-Based Person Authentication, 177-183, 1997.

- [70] J. Steffens, E. Elagin, and H. Neven, "PersonSpotter – fast and robust system for human detection, tracking and recognition," In Proceedings, International Conference on Automatic Face and Gesture Recognition, 516-521, 1998.
- [71] J. Bigun, B. Duc, F. Smeraldi, S. Fischer, and A. Makarov, "Multi-modal person authentication," In Face Recognition: From Theory to Applications, H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang, Eds. Springer-Verlag, Berlin, Germany, 26-50, 1998.
- [72] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," Computer Vision Image Understanding, vol. 91, 214-245, 2003.
- [73] S. Gong, S. McKenna, and A. Psarrou, "Dynamic Vision: From Images to Face Recognition," World Scientific, Singapore, 2000.
- [74] L. Klasen, and H. Li, "Faceless identification," In Face Recognition: From Theory to Applications, H. Wechsler, P.J. Phillips,  V. Bruce, F.F. Soulie, and T.S. Huang, Eds. Springer-Verlag, Berlin, Germany, 513-527, 1998.
- [75] H. Rowley, S. Baluja, and T. Kanade, "Neural network based face detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, 1998.
- [76] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models – their training and application," Computer Vision and Image Understanding, vol. 61, 18-23, 1995.
- [77] J. Liu, and R. Chen, "Sequential Monte Carlo methods for dynamic systems," Journal Am. Stat. Assoc., vol. 93, 1031-1041, 1998.
- [78] M. Isard, and A. Blake, "Contour tracking by stochastic propagation of conditional density," In Proceedings, European Conference on Computer Vision, 1996.

- [79] B. D. Zarit, B. J. Super, and F. K. H. Quek, "Comparison of five color models in skin pixel classification," In ICCV'99 Int'l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems, 1999.
- [80] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, May 2002.
- [81] S. H. Lin, S. Y. Kung, and L. J. Lin, "Face recognition/detection by probabilistic decision-based neural network," IEEE Trans. Neural Network, vol. 8, 114-132, 1997.
- [82] K. Okada, J. Steffans, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. von der Malsburg, "The Bochum/USC Face Recognition System and how it fared in the FERET Phase III Test," In Face Recognition: From Theory to Applications, H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulie, and T. S. Huang, Eds. Springer-Verlag, Berlin, Germany, 186-205, 1998.
- [83] P. Penev, and J. Atick, "Local feature analysis: A general statistical theory for object representation," Netw.: Computat. Neural Syst., vol. 7, 477-500, 1996.
- [84] L. Wiskott, J.-M. Fellous, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, 775-779, 1997.
- [85] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic face identification system using flexible appearance models," Image Vis. Comput., vol. 13, 393-401, 1995.
- [86] Da Silva, S., and Agbinya, J.I., "Face Recognition Programming on Mobile Handsets", To appear in the Proceedings of ICT 2005 - 12th International Conference on Telecommunications, Cape Town, South Africa, 3 – 6 May 2005.
- [87] <http://ise.stanford.edu/2003projects/ee368/Project/reports/ee368group03.pdf>

- [88] A. Albiol, L. Torres, C.A. Bouman, and E.J. Delp, "A simple and efficient face detection algorithm for video database applications," in Proceedings of the IEEE International Conference on Image Processing, Vancouver, Canada, September 2000, vol. 2, pp. 239-242.
- [89] C. Garcia and, M. Delakis, "A Neural Architecture for Fast and Robust Face Detection", (IEEE-IAPR International Conference on Pattern Recognition (ICPR2002), Québec City, August 2002, p. 40-43.
- [90] H. Chang, and U. Robes, May 2000, "Face detection", <http://www-cs-students.stanford.edu/~robles/ee368/main.html>
- [91] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, May 2002.
- [92] A. Bertran, Y. Huanzhou, and P. Sacchetto, Spring 2002, "Face Detection Project Report", <http://ise.stanford.edu/2002projects/ee368/Project/reports/ee368group17.pdf>
- [93] Stan Birchfield. "Elliptical Head Tracking Using Intensity Gradients and Color Histograms." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 232-237, June 1998.
- [94] F. Hausdorff. "Set Theory". New York: Chelsea Publishing Company, third edition, 1978.
- [95] J. I. Agbinya, and D. Rees, "Multi-Object Tracking in Video", Real-Time Imaging, Academic Press, vol. 5, iss. 5, pp. 295-304, October 1999.
- [96] Agbinya J.I., Lok B., Da Silva S., and Wong A., "Automatic Online Porn Detection and Tracking", To appear the Proceedings of ICT 2005 - 12th International Conference on Telecommunications, Cape Town, South Africa, 3 – 6 May 2005.