# *Towards an efficient, unsupervised and automatic face detection system for unconstrained environments*

Department of Electrical and Electronic Engineering

Loughborough University

# Towards An Efficient, Unsupervised and Automatic Face Detection System for Unconstrained Environments

**Lihui Chen**

A thesis submitted in partial fulfillment for the award of

Doctor of Philosophy at Loughborough University

March 2006

Supervisor: Christos Grecos, PhD MIEEE

Research Director: Prof. David Parish

Applied Signal Processing Group

# Acknowledgement

First of all, I would like to express my gratitude and appreciation to my supervisor, Dr. Christos Grecos, for his precious advice and valuable instruction in the process of my research. Under his direction, I started with a thorough understanding of my research area. I have great respect for his kind attention and time.

A special note of thanks should be recorded for my colleague, Mr. Mingyuan Yang. In our discussions, he had provided valuable information and help. With that, I am able to go deeper into my research and accomplish more.

Secondly, I would to add my grateful thanks to my family members for their support and care throughout the years.

Also, I would like to extend my thanks to the staff of the Department of Electronic and Electrical Engineering for their assistance in my research, and to the staff of the Pilkington Library of Loughborough University for their help in searching for useful information.

Last but not least, I should like to acknowledge the support of all my friends.

# Abstract

Nowadays, there is growing interest in face detection applications for unconstrained environments. The increasing need for public security and national security motivated our research on the automatic face detection system. For public security surveillance applications, the face detection system must be able to cope with unconstrained environments, which includes cluttered background and complicated illuminations. Supervised approaches give very good results on constrained environments, but when it comes to unconstrained environments, even obtaining all the training samples needed is sometimes impractical. The limitation of supervised approaches impels us to turn to unsupervised approaches.

In this thesis, we present an efficient and unsupervised face detection system, which is feature and configuration based. It combines geometric feature detection and local appearance feature extraction to increase stability and performance of the detection process. It also contains a novel adaptive lighting compensation approach to normalize the complicated illumination in real life environments. We aim to develop a system that has as few assumptions as possible from the very beginning, is robust and exploits accuracy/complexity trade-offs as much as possible. Although our attempt is ambitious for such an ill posed problem-we manage to tackle it in the end with very few assumptions.

***Keywords:*** face detection, skin color, configurational verification, morphological operators, eye detection, local gray value, mouth area, corner detection

# Contents

# List of Figures

# List of Tables

CHAPTER 1

# Introduction and Summary

## 1.1 Introduction

Each human has a face that is unique and provides information about the identity of its owner.

### 1.1.1 Human Face Recognition

We detect and recognize human faces in our daily life, effortlessly. The face recognition skill, also named face perception skill, seems to take no time for us to learn. Indeed, by two months of age, face perception skill has already been developed, so specific areas of the brain are known to be activated by viewing faces [1]. According to Roark et al [2], humans can keep track of hundreds, if not thousands, of individual faces, which far exceeds our ability to memorize individual exemplars from any other class of objects. This human face 'processing' skill can make simultaneous use of a variety of information from the face, including information about the age, sex, race, identity and even current mood of the person. We are further able to track facial motions that alter the configuration of features, making it difficult to encode the structure of the face. What is more, we are capable of these impressive feats of visual information processing even when the viewing conditions are variable or less than optimal.

One of the most widely accepted theories of face recognition [3] argues that understanding faces involves several stages; from basic perceptual manipulations on the sensory information to derive details about the person (such as age, gender or attractiveness), to being able to recall meaningful details such as their name and any relevant past experiences of the individual.

The model and procedure of human face recognition remain an arguing topic. Many models have been built, some are even contradictory to each other. An

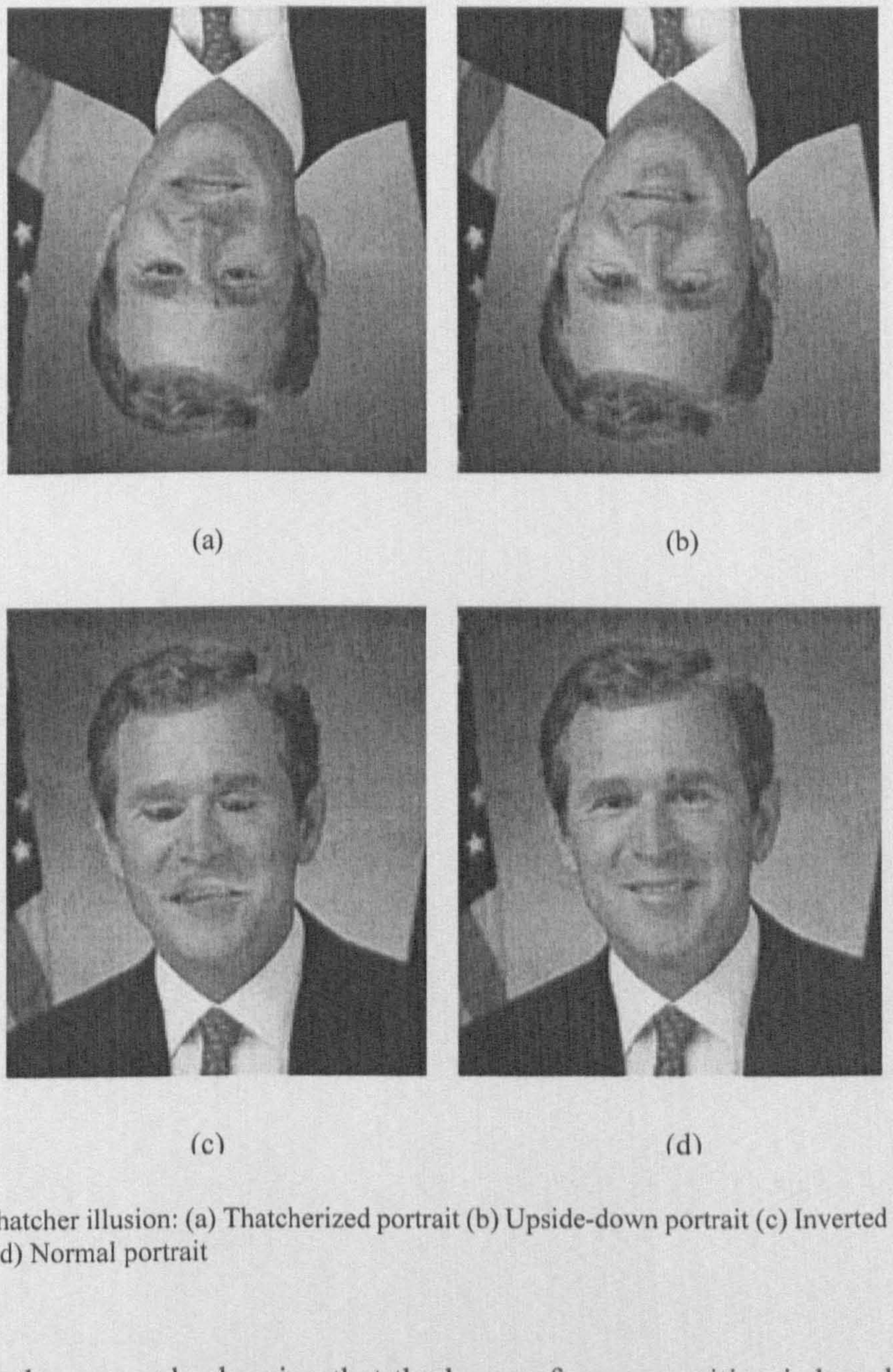
outstanding model developed by psychologists Vicki Bruce and Andrew Young [3] argues that face perception might involve several independent sub-processes working in unison.

- A 'view centered description' is derived from the perceptual input. Simple physical aspects of the face are used to work out age, gender or simple facial expressions. Most analysis at this stage is on feature-by-feature basis.
- This initial information is used to create a structural model of the face, which allows it to be compared to other faces in memory, and across views. This explains why that the same person seen from a novel angle can still be recognized. This structural encoding can be seen to be specific for upright faces as demonstrated by the Thatcher effect.
- The structurally encoded representation is transferred to notional 'face recognition units' which in conjunction with 'person identity nodes' allow the person to be identified by information from semantic memory. Interestingly, the ability to produce someone's name when presented with their face has been shown to be selectively damaged in some cases of brain injury, suggesting that naming may be a separate process from being able to produce other information about a person.

This model suggests that the human face recognition process is based on a feature-by-feature architecture. Also, data from memory experiments suggest that humans use both feature-based and configured information to recognize faces [4], with perhaps special reliance on facial configurations [5]. Manipulations aimed at perturbing the configuration of a face or at disrupting our ability to process its configurational information are tested [4], [6], [7], and all of them turn out to strongly affect the human recognition accuracy and processing speed.

A famous example of these manipulations is the Margaret Thatcher Illusion [8]. A picture of a face can be 'Thatcherized' by inverting the eyes and the mouth, then inverting the entire picture, as shown in Fig. 1.1(a). Most people do not notice the gross distortion of the configuration of the facial features in the Thatcherized portrait, while it is easy to everyone to figure it out in Fig. 1.1(c). There is the evidence that

humans are highly sensitive to the configuration of the features in a face, but the Margaret Thatcher illusion illustrates that even human face recognition has some important processing limits for non-typical views.



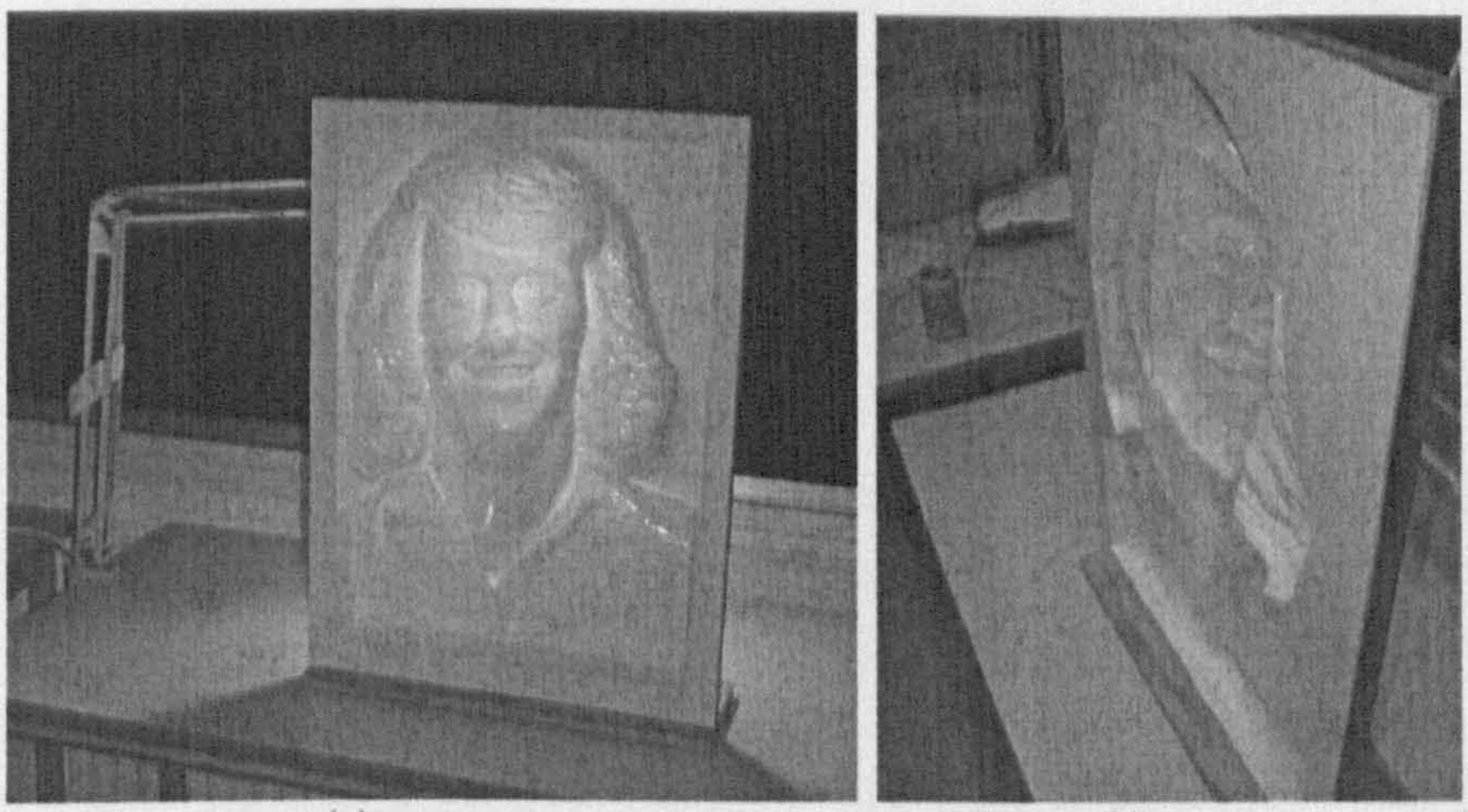
(a) (b)

(c) (d)

*Fig 1.1* Thatcher illusion: (a) Thatcherized portrait (b) Upside-down portrait (c) Inverted Thatcherized portrait (d) Normal portrait

Another example showing that the human face recognition is based on feature and configuration is the Hollow Face illusion, as shown in Fig. 1.2. The hollow face illusion illustrates that human face recognition in is not based on 3-D shape or convexity of the face, but the configuration of facial features and the properties of the

features. This provides an important cue that face detection and recognition in still images is an achievable goal.



(a) (b)

*Fig 1.2* Hollow face illusion. (a) Illusion (b) Real convexity

Furthermore, it is possible that the facial movements may also complicate the job of the perceiver for recognizing faces. It is likely that the information needed to recognize faces can be found in the invariant or unique form and configuration of the features. Non-rigid facial movements can alter the configuration of facial features, also in a dramatic way. Anyway, research on the effects of various motions on recognition accuracy is just beginning [9].

### 1.1.2 Computer Face Recognition

Computer face recognition is an important topic of machine vision and learning. Research in computer face recognition is motivated not only by the fundamental challenges this recognition problem poses, but also by numerous practical application where human identification is needed. Comparing with other biometric technologies, face recognition is natural, non-intrusive and easy to use. Thomas Huang et al [11] summarize the applications of computer face recognition into 10 major categories, as listed in Table 1.1. A typical face recognition system consists of four modules: detection, alignment, feature extraction and matching.

Table 1.1

FACE RECOGNITION APPLICATION CATEGORIES

| Category | Exemplar application scenarios |
|---|---|
| Face ID | Driver licenses, entitlement programs, immigration, national ID, passports, voter registration, welfare registration |
| Access control | Border-crossing control, facility access, vehicle access, smart kiosk and ATM, computer access, computer program access, computer network access, online program access, online transaction access, long distance learning access, online examinations access, online database access |
| Security | Terrorist alert, secure flight boarding systems, stadium audience scanning, computer security, computer application security, database security, file encryption, intranet security, Internet security, medical records, secure trading terminals |
| Surveillance | Advanced video surveillance, nuclear plant surveillance, park surveillance, neighborhood watch, power grid surveillance, CCTV control, portal control |
| Smart cards | Stored value security, user authentication |
| Law enforcement | Crime stopping and suspect alert, shoplifter recognition, suspect tracking and investigation, suspect background check, identifying cheats and casino undesirables, post-event analysis, welfare fraud, criminal face retrieval and recognition |
| Face databases | Face indexing and retrieval, automatic face labeling, face classification |
| Multimedia management | Face-based search, face-based video segmentation and summarization, event detection |
| HCI<br>Human computer interaction | Interactive gaming, proactive gaming |
| Others | Antique photo verification, very low bit-rate image & video transmission, etc. |

In the marketing perspective, face recognition systems have a broad range of applications, including some critical ones. For example, face ID in security systems is becoming more and more important for public security under the threat of terrorisms. Also, face recognition and description is a key feature for video indexing, searching and management. Thus, the potential demand for face recognition systems is very high. If a mature face recognition system appears, the market would rapidly develop.

In the paper of [11], it is pointed out that at present the face recognition technology is most promising for small or medium scale applications, such as office access control and computer log in. And it still faces great technical challenges for large-scale deployments such as airport security and general surveillance. Current face recognition technology is still not robust, especially in unconstrained environments, and recognition accuracy is not enough. Several systems installed at public places have not received positive feedback based on their poor performance. But nowadays, under the shadow of global terrorism, public security is at the risk. Thus requirement for a full-automatic, or at least semi-automatic, public surveillance system is increasing. Such a system must also be able to work in outdoor and public environment, which makes it a very challenging system to implement.

In [10], Stan Z. Li summarizes the technical challenges of face recognition technology as below:

- Large variability in facial appearance
- Highly complex nonlinear manifolds
- High dimensionality and small sample size

Li suggests two strategies for dealing with the above difficulties: feature extraction and pattern classification based on the extracted features. The first includes two levels of processing: (1) normalize face images geometrically and photometrically; and (2) extract features in the normalized images which are stable with respect to such variations. The second strategy is to construct classification engines able to solve difficult nonlinear classification and regression problems in the

feature space and to generalize better.

We know that trying to build up a universal face recognition system is impractical. So to build up a system, one has to setup some constraints. Like what the intended situation for the application and how strong constraints are assumed, including pose, illumination, facial expression, age, occlusion, and facial hair. The current maturity of the technology is as following: Real-time face detection and tracking in the normal indoor environments is relatively well solved, whereas more work is needed for handling outdoor scenes. Assuming the image resolution is good enough, current face recognition technologies work well for frontal faces without exaggerated expressions and under illumination without much shadow. Recognizing faces in an unconstrained daily life environment without user's cooperation remains a challenge.

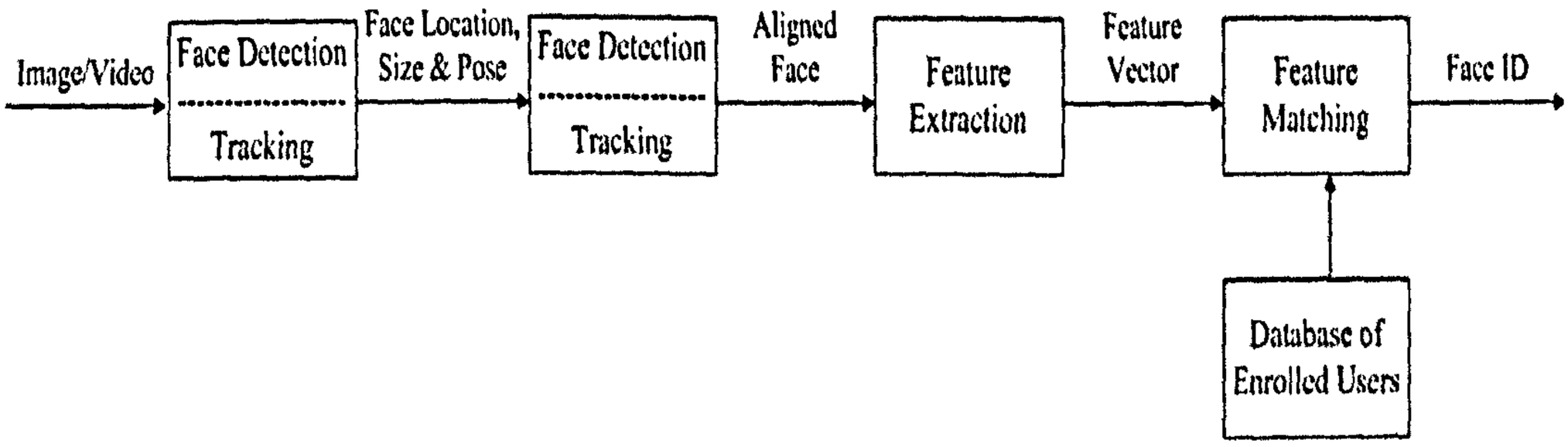



*Fig 1.3* Face Recognition processing flow

### 1.1.3 Computer Face Detection

Face Detection is defined in [12] as: Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face.

Face detection is a key problem in pattern recognition. It is important because it is not only an essential first step towards a fully automated face recognition system, as shown in Fig. 1.3, but also its reliability has a major influence on the performance and the usability of the entire face recognition system. According to Stan Z. Li [10], an ideal face detector should be able to identify and locate all the present faces

regardless of their position, scale, orientation, age, and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions, complex backgrounds in real life and the image/video representations. M.H. Yang et al [12] attributed the challenges associated with face detection to following factors:

- **Pose.** The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.
- **Presence or absence of structural components.** Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, color, and size.
- **Facial expression.** The appearances of faces are directly affected by a person's facial expression.
- **Occlusion.** Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.
- **Image orientation.** Face images directly vary for different rotations about the camera's optical axis.
- **Imaging conditions.** When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

In Fig. 1.4, we give out an example for each of the factors.

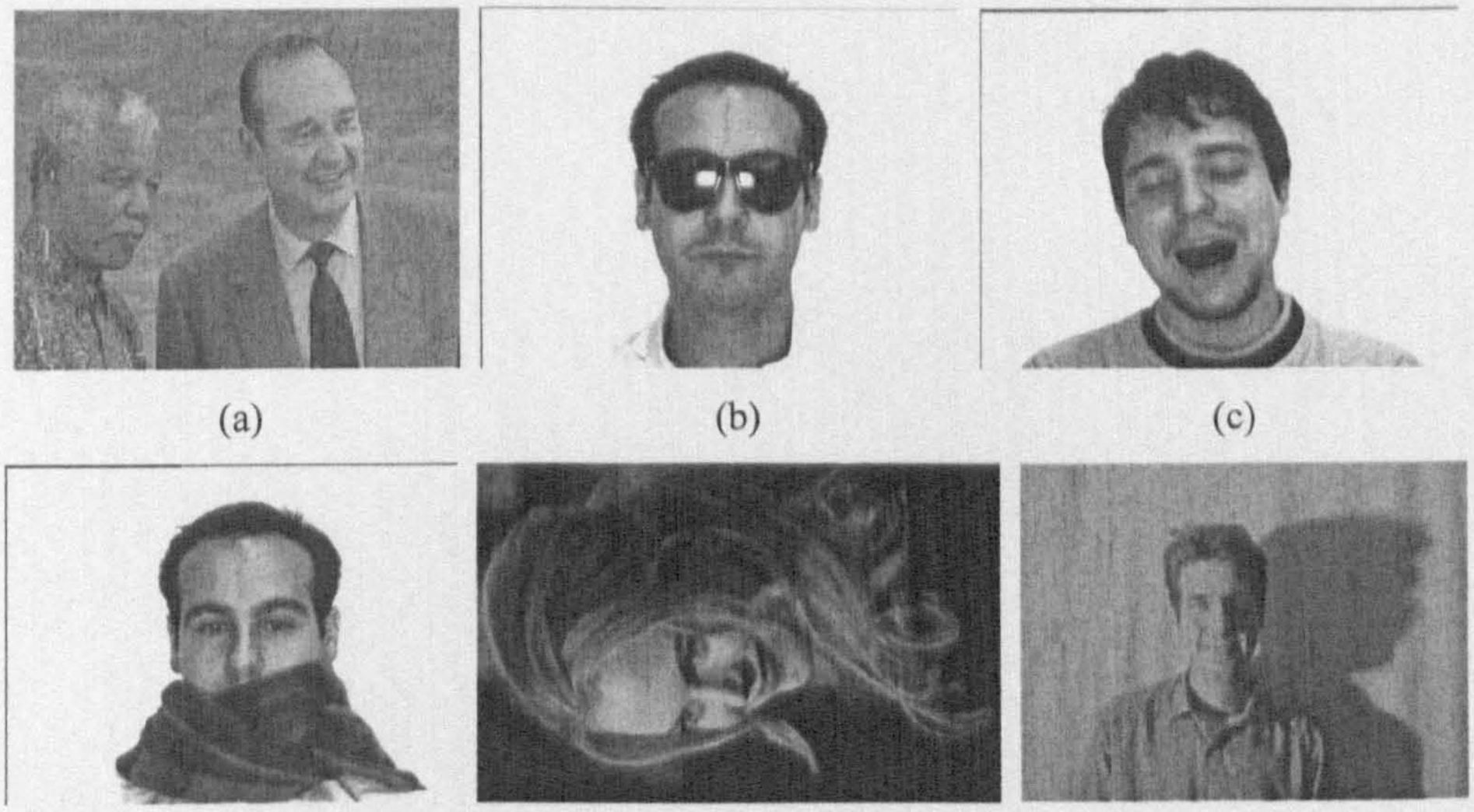



*Fig 1.4* Face detection challenges: (a) pose (b) present of structural elements (c) facial expression (d) occlusion (e) image orientation (f) image conditions

With considering the aforementioned psychological research results, we decided that our face detection system should be concentrating on with the first strategy, which is feature extraction. With a robust feature detector, the face / non-face classification will not be a big problem. If the feature detector is weak, given the system would be used in the unconstrained environment, it would be very difficult for the classification engine to do the classification correctly.

## 1.2 Summary of this thesis

In this thesis I restrict myself to the face detection problem, towards an efficient and unsupervised face detection system.

### 1.2.1 Chapter 2

Chapter 2 is a literature review of existing face detection approaches. In the chapter, we will discuss and compare various approaches to the face detection problem, including supervised approaches. The comparison will enable us to better understand the assumptions, advantages and limitations of these approaches. At the

same time, we explain the reason that we choose the architecture of our scheme, and how that helps to achieve our final goal of the face detection system we are after.

In the following two chapters, I would represent my scheme towards the efficient and unsupervised face detection system. These chapters will form the main body of this thesis.

### 1.2.2 Chapter 3

In the chapter, a skin region detector is represented. The detector starts with a novel adaptive lighting compensation algorithm, which is proved to be fast yet effective. Then a skin color model is developed. It is not a model tuned for some specific image databases, but just a general model setup from a few sample images gathered from real life. Finally, it comes to the most important step of our skin color detector. Morphological operators are applied to the mask generated by skin color filter, which substitutes the traditional merging process used in [18].

Experiments results are presented and comparisons with several existing approaches are made. Comparisons would be able to demonstrate that our scheme is faster and more effective on skin color region detection. Also, the accuracy of our detector on the selected face databases would be consistent.

### 1.2.3 Chapter 4

Based on the result of the skin region detector, multiple cues are used to detect the eyes. First of all, an eye detector utilizing corners, local gray pixels and edges is developed. Then with the possible eye candidates detected, we start our face verification process with the configurational knowledge of human faces, which that the mouth would be within a triangular range of the two eyes. The combination of feature extraction and configuration verification is inspired by the psychological research results in human face perception.

### 1.2.4 Chapter 5

This brings us to the end of this thesis. The conclusion chapter would summarize the contribution of this thesis in terms of its investigations into various approaches on

face detection, and building up an efficient and unsupervised face detection system based on features and configuration in the end, together with the novel approaches and heuristics presented in it. The chapter would also show possibilities for extension and improvements of the work and affect on other related works.

CHAPTER 2

# Literature Review

## 2.1 Introduction

Face detection has been studied for decades due to it is the foundation for many applications, such as face recognition, human-computer interaction and security surveillance systems.

Various approaches for face detection are discussed in detail in [12]. The paper also categorizes the current face detection methods into four major categories, as shown in Table 2.1.

Table 2.1

CATEGORIZATION OF METHODS FOR FACE DETECTION IN SINGLE IMAGE

| Approach | Representative Works |
|---|---|
| Knowledge-based | |
| | Multi-resolution rule-based method [23] |
| Feature Invariant | |
| • Facial feature | Grouping of Edges [24][25] |
| • Texture | Space Gray-Level Dependence matrix (SGLD) [26] |
| • Skin-color | Mixture of Gaussians [27][28] |
| • Multiple features | Integration of skin color, size and shape [29] |
| Template Matching | |
| • Predefined face templates | Shape template [30] |
| • Deformable templates | Active Shape Model (ASM) [31] |
| Appearance-based methods | |
| • Eigenface | Eigenvector decomposition and clustering [32] |
| • Distribution-based | Gaussian distribution and multilayer perception [33] |
| • Neural network | Ensemble of neural networks and arbitrary schemes [34] |
| • Support Vector Machine (SVM) | SVM with polynomial kernel [35] |
| • Naïve Baysian classifier | Joint statistics of local appearance and position [36] |
| • Hidden Markov Model (HMM) | Higher order statistics with HMM [37] |
| • Information-Theoretical approach | Kullback relative information [38][39] |

Appearance-based approaches, such as the ones utilizing Neural Networks [13] or view-based approaches [14], need numerous positive and negative samples as training samples. As pointed out in the survey on statistical pattern recognition by Jain et al [15], the performance of a classifier depends on both the number of available samples as well as the specific value of the samples. As to the template matching approaches, it is mostly depended on shapes. A robust shape descriptor is needed to extracted required shape features for the match to succeed. Yang et al [12] provides much more detail description and comparison of the approaches listed in Table 2.1.

At the same time, the goal of designing a recognition system is to classify future test samples which are likely to be different from the training samples. Therefore, optimizing a classifier to maximize its performance on the training set may not always result in the desired performance on a test set. In essence, in spite of the seemingly different underlying principles, most of the well-known neural network models are implicitly equivalent or similar to classical statistical pattern recognition methods. What is more, the average computation cost of these techniques is high. On the other hand, template matching approaches [16], [17] are usually applied to frontal face detection and recognition. While computationally cheaper, they are sensitive to factors like facial expression, illumination, scale and orientation.

One of the most successful and well-studied techniques to face recognition is the appearance-based method. The most popular appearance-based approach for face detection might be the Eigenface approach, which is based on PCA. PCA is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original n-dimensional data onto the k<<n-dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pair-wise de-correlated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. Turk and Pentland [32] use Principal Component Analysis to describe face images in terms of a set of basis

functions, or "eigenfaces."

Another famous appearance-based approach is the Linear Discriminant Analysis (LDA) [40], which is a supervised algorithm. LDA searches for the project axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other. Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linearly separable space using bases that are not necessarily orthogonal. It is generally believed that algorithms based on LDA are superior to those based on PCA. However, some recent work [41] shows that, when the training data set is small, PCA can outperform LDA, and also that PCA is less sensitive to different training data sets.

In [15], as a conclusion, it is pointed out that no single approach for classification is "optimal" and multiple methods and approaches have to be used. According to Hsu et al [18], face detection algorithms using holistic representations have the advantage of finding small faces or faces in poor-quality images, while those using geometrical facial features provide a good solution for detecting faces in different poses. A combination of holistic and feature-based approaches [19], [22], is a promising avenue for both face detection and face recognition.

## 2.2 Skin Color Detection

### 2.2.1 Introduction

Color information has been widely used to assist face detection. Comparing to the shape descriptor and other feature descriptors, color is a low level cue that can be discriminative and computationally fast. A typical geometric change, such as rotation, translation or scaling can hardly affect the color information. It is also well-known that we humans tend to spot easily color changes in the skin tones [40]. At the same time, its limitation is also well-known. Color is very sensitive to illumination changes and the color bias of the camera. And commonly, information

like the white balance setting and the prevailing illumination condition detected by camera when the image was captured can not be assumed. Also, using only color for face detection is not enough, since it cannot necessarily distinguish faces from other objects with a similar appearance like body skin and wood. Therefore, other cues are needed to be combined with color for the verification of the skin color region as a face. But still, color is useful in the pre-processing step because it may significantly remove many pixels which are obvious belonging to non-face objects, which in turn alleviate the computation cost of upcoming steps.

So an essential part of a face detection system is the modeling of the skin-tone color distribution and the design of effective skin color filters [43], [44], [45]. Such skin filters can rapidly remove non-skin color pixels, which in turn dramatically reduce the amount of pixels needed to be processed in subsequent steps of the recognition process. Unfortunately, all the aforementioned color-based face detection approaches will meet difficulties for images with complex background and illuminations. They can even inflict potential loss of facial features, thus degrading performance in the later face recognition stages.

### 2.2.2 Skin Color in Different Color Spaces

Numerous color spaces are developed for various applications [46]. Most of them are transformed from the RGB color space and many of them are trying to split the color information into chromaticity and intensity components, such as the $YC_bC_r$ and *HSV* color spaces. However, the de-convolution of luminance and chromacity planes is not necessarily successful in all cases. In some cases, the chromaticity component may still depend on the intensity of the RGB values.

Several studies have been made to evaluate the color spaces for skin detection [47][48][49][50][51][55][56]. In [51], Jones and Rehg demonstrated with their large-scale experiment that in RGB space the histogram provides slightly better results than the mixture of Gaussians. Zarit et al [56] compared five color spaces (*CIE Lab*, Fleck *HS*, *HSV*, normalized *RGB* and $YC_bC_r$). According to their study, the look-up table approach, which decides if the color is skin-tone by looking up a table of possibilities of skin-tone for the full spectrum, performs best with

HS-spaces. While for the Baysian decision-based approaches in skin detection, the choice of color spaces does not matter. Terrillon et al [53] compared the performance of nine color spaces (*TSL*, *NCC*, *RGB*, *CIE xy*, *CIE SH*, *HSV*, *YIQ*, *YES*, *CIE Luv*, *CIE Lab*) on a single Gaussian function and Gaussian mixture in their skin color modeling and found out that the single Gaussian model provided the best results in normalized color spaces. Caeteno et al [48] compared both the models Terrillon used in normalized rg coordinates using skin pixels extracted from face, arms and leg areas of different ethnic groups and found the two models have similar results. Yang et al [57] compared the Gaussian mixtures with self-organizing maps (SOMs) in four color spaces (*HSV*, Cartesian *HS*, *TSL* and normalized *rg*), the Gaussian mixtures only result better in normalized *rg* space, while the SOM approach produces a consistent detection rate in all the four color spaces. In [55], Shin et al evaluated eight color spaces (normalized *RGB*, *CIE*, *XYZ*, *CIE Lab*, *HIS*, Spherical coordinate transformation, $YC_bC_r$, *YIQ* and *YUV*) with criteria: (1) separability between skin and non-skin classes (2) compactness of skin colors (3) robustness against illumination variations. The conclusion is that the *RGB* space is the most preferable and the illumination component increases separately when one can assume the object has a limited intensity range under the illumination or having enough data to assess reliably all possible intensity cases.

The previous comparative studies indicate that the normalized RGB space should be suitable for many applications using skin color, although we understand that the results depend on the different test images and different experiment parameters.

### 2.2.3 Skin Color Models

The major factor needed to be considered before employing a skin color model is illumination. If knowledge is available about possible prevailing and calibration lighting, it can be used to choosing or tuning the color model. On the other hand, a pure skin tone model fails when the colors of the object shift owing to the illumination variation in time or in the spatial domain.

There are many ways to establish the skin color model. We can sort them into

several major categories:

- **Statistics Based Classification.** Many statistical techniques have been employed for skin detection: parametric, such as a Gaussian or Gaussian mixtures [58]; semi-parametric, such as the Self-Organizing Map [59] and neural networks [60]; and non-parametric, such as a histogram [61] (although the number of bins selected would influence the outcome, it is generally called non-parametric).

- **Selection Based on a Region in a Color Space.** A region in the color space is to be determined from an image set considered to represent varying skin appearance. After the skin tone region is defined, it can be used to filter out non-skin pixels. The region can be represented in two ways: a look-up table or threshold values [62]. A problem related to these approaches would be the boundary decision for the skin region. Outliners and values at the boundary are likely to be noises, and should be removed.

- **Color Correction for a Region-based Approach.** Hsu [18] suggested a skin color model with a color correction step. But their color correction approach has invalid assumptions that would cause data loss [50][63]. We will explain the problems of their assumptions in detail when it is compared with our lighting compensation approach. Since none of the color constancy approaches is robust enough, we do not consider using it when setting up our own skin color model.

- **Skin Color Modeling with a Skin Locus.** The goal of skin locus [63] is to provide robustness against changing intensity and tolerance toward varying illumination chromaticity. The knowledge of the so-called 'skin-locus' itself does not assume the distribution or probabilities in the spectrum for the skin color. It only defines the area that has the maximum likelihood for possible chromaticity and in this way tolerates inherently non-uniform illumination fields. For this approach, extra information like spectral power distribution of illuminations or possible white balancing conditions must be known. It is shown to be better under lighting variations than other

state-of-the-art methods [66], and fails in some cases [63][64]. To calculate the locus, one needs spectral knowledge of the camera, illumination conditions and skin. Although the spectral knowledge of the illumination and skin can be easily obtained [49][67], the camera characteristics are often unavailable. The information can be obtained either from the manufacturer of the camera, or by taking face images in the illumination range with the allowed camera settings. For the latter case, the illumination range must be based on the given application. Examples to be used should contain the maximum and minimum color temperature, and some from the middle range. Also, one has to know the number of possible white balancing and the lighting conditions under which they are done. Skin locus seems to be a promising approach to gain a hardware-optimized result to handle the chromaticity change for skin color, and can be used in the architecture we suggest. But since it is hardware related, therefore, we are not going to discuss it in this thesis.

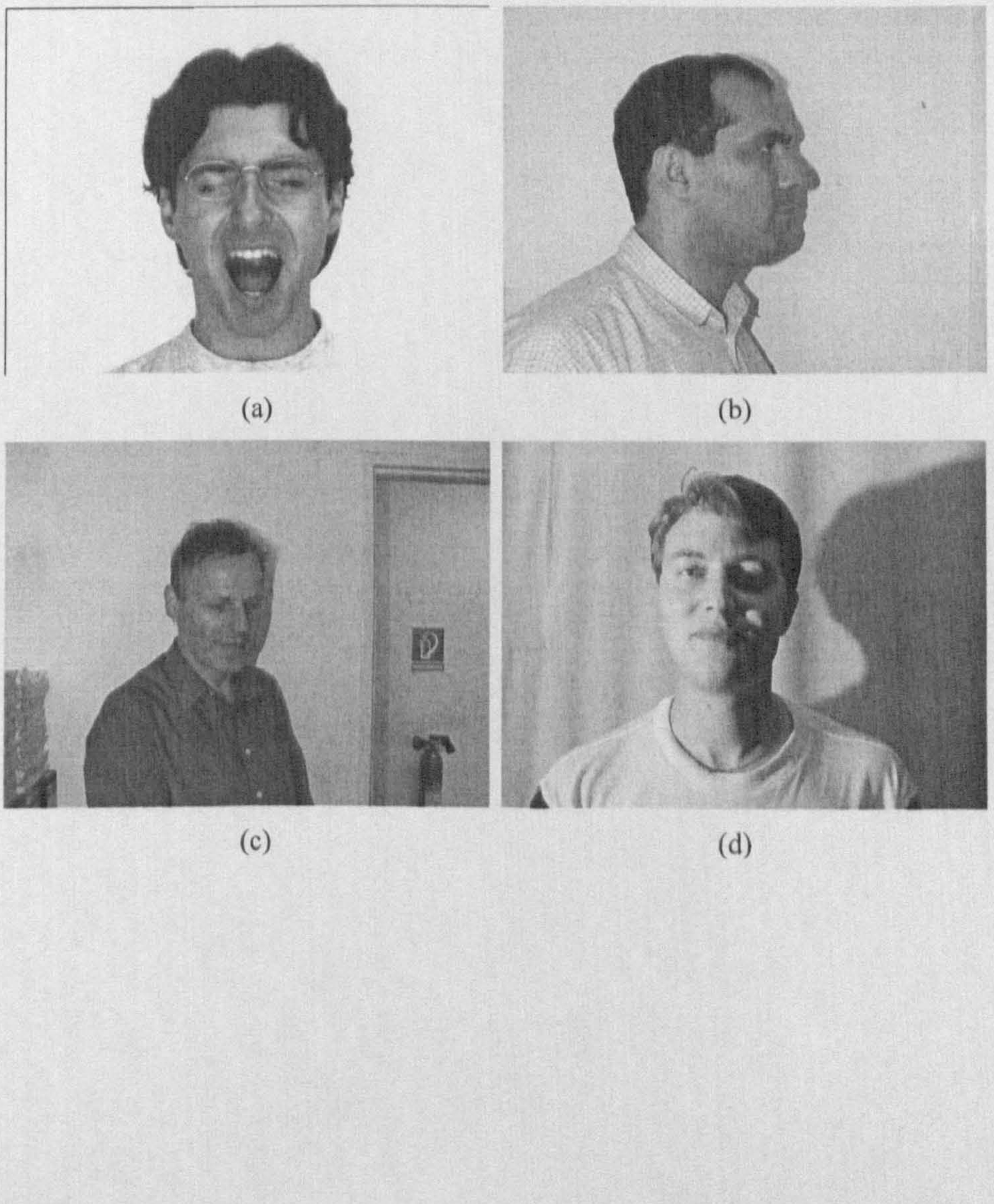
## 2.3 Facial Feature Detection

Facial feature detection has attracted a lot of attention due to its applications in fields of computer vision and graphics and a lot of research has been done to capture the communicative ability of the face. Applications like facial expression analysis, animation and coding need to detect the facial features robustly and efficiently. Also, rather then just detecting the positions of facial features, the shape information would be needed as well. The variability in appearance of facial features changes, due to pose, lighting, facial expressions etc., makes the task difficult and complex.

In a typical face region, many features are considered for face detection, such as the eyes and/or their corners, the nostrils, and the mouth and/or their corners [96][97]. Eye detection is a key problem in both face detection and face recognition because eyes are important facial features of human beings. It is known that face alignment, which has a large impact on recognition accuracy [80], [81], is usually

performed with the use of eye positions [82]. It is also important for applications such as iris identification, human-computer interaction, face tracking.

According to Ji et al [83], current eye detection methods fall into two categories: active and passive. Active eye detection generally uses the special physical property of pupil under Infrared Radiance (IR) illumination [84]. These methods have been proved to be robust and accurate for specific applications. But the limitations of these methods are obvious. On one hand, IR illumination would be interfered in outdoor environments where there are other IR illumination sources, i.e. the strong reflection of the full spectrum sun lights. On the other hand, these methods cannot be used to process ordinary images or videos which do not contain the IR information when captured. What is more, sometimes even the bright pupil effect needed for IR-based eye detectors would be affected by pose, orientation and shadow. Examples of these cases are given in Fig 2.1. Given all the limitations mentioned, the active IR approach needs a controlled environment of usage, which is obviously not suitable for the face detection system we are after.



(a) (b) (c) (d)

*Fig 2.1* Pupil lost cases. (a).eye closed (b) profile (c) looking down (d) shadow

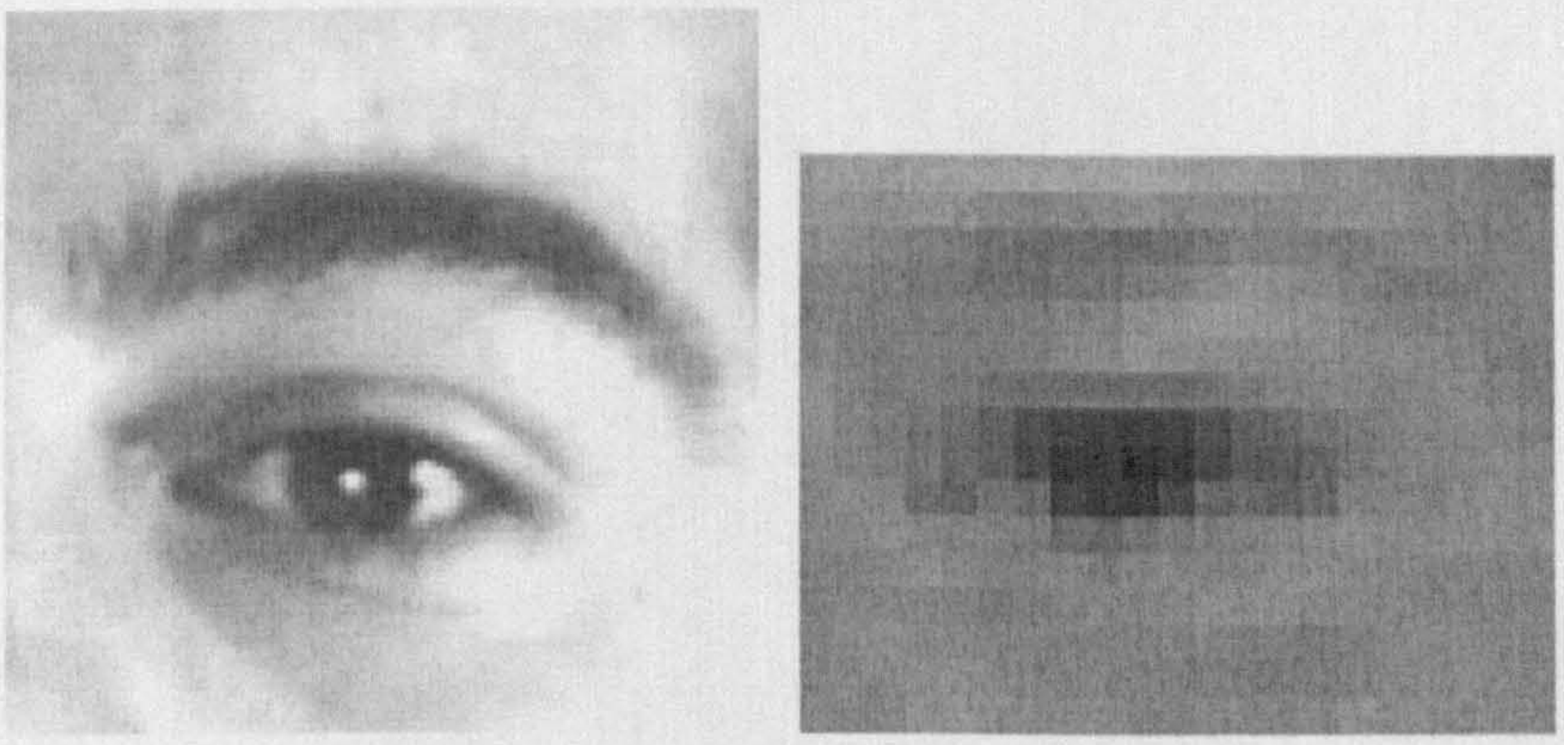
Passive eye detection methods detect eyes within visual spectrum. For passive eye detectors, Huang and Wechsler [85] divided these detectors into two major categories. The holistic approach attempts to locate the eyes using global representations [89], like Pentland et al's modular eigenspaces [86] and Sarmaria's HMM based algorithm [87]. The abstractive approach extracts and measures discrete local features, and then employs standard pattern recognition techniques to locate the eyes using these features. Features like gradient directions [88], projection function [89], and deformable templates [90], [91] are used in previous works.

Most of these previous works concentrated on gray level images, which features only depend on gray intensity. While color images are becoming popular, we can have more cues, but at the down side, the data volume to be processed increases dramatically. Thus, we need to find some efficient cues for fast eye detection.

In our system, our facial feature detection process starts with eye corner detection. There are several reasons choosing to start with the eye corner detector.

First of all, it is well known that each eye has two corners. Whatever perspective the face image is, i.e. the profile angle as Fig 2.1(b), at least one eye corner would still exist. The only case eye corners all disappear happens when the eyes are widely open due to expressions like astonishment. Even in this case, eyebrows and pupils can still help forming corners detectable. Therefore, eye corners can be considered as both pose and expression invariant.

What is more, corner detectors can also be invariant to rotation, scale and small distortions. As shown in Fig 2.2, detectable eye corners can be formed in both high resolution and low resolution images.



(a) (b)

*Fig 2.2* Eyes in different resolutions (a) High resolution (b) Low resolution

Eye corners are also robust to illumination changes due to its distinctiveness from the adjacent skin, the convexity of the eyeball, the concavity of the eye sockets, and the wrinkles of the eye lids. As shown in Fig 2.3, the right eye is in shadow of the nose bridge. Details of eyeballs and eye lids are missing. Texture and color distinctiveness is also severely weakened. Yet the convexity of the eyeball and the wrinkles on the eye lid still manage to form corners.

*Fig 2.3* Eyes in different illumination conditions

For the reasons above, we consider corner a robust feature to start with our eye localization process. At the same time, facial hair and shadows on the face would also form corners, which will cause false positives. Therefore, after the eye corners are detected, more heuristics would be needed to verify the eye.

As we understand that facial feature detectors will not be perfect, false positives

would be inevitable. Therefore, after the facial feature detection, an additional verification process would be necessary to localize the actual face.

CHAPTER 3

# Skin Region Detector

## 3.1 System Design

First of all, we would dedicate a section to describe our system architecture briefly.

After all the analysis in the previous chapter, we choose to setup our face detection system with the architecture shown in Fig. 3.1. It is similar to the architecture suggested by Hsu et al in [18].

First, a lighting compensation is needed to calibrate the skin color. Then a skin color filter comes into place to eliminate the obvious non-face pixels. After that, the mask generated by the skin color filter is refined by using morphological operators to remove noise and regain the important facial features. Now with the skin patches, facial features as eyes would be detected. When it is done, mouth area and human face configuration information would be utilized to locate the faces and eliminate false positive results.

This is the theoretical system design. In a real system implementation, the current lighting compensation approach might need to be replaced with algorithms that are optimized for image capture devices. If the lighting compensation algorithm is changed, the skin color model will also need to be changed. Also, images might be changed into video frames for video surveillance and tracking applications.

Dependent on the requirement of the application, an optional step of face contour extraction might be added to the architecture. The process would extract the contour of the human face by some edge detection/tracing algorithms. For example, the Hough transformation [19] that Hsu used in [18] or the JPEG-LS edge detector [20] may be used. Then it will crop the face base on the contour extracted

and feed it for later process. This step is not included in this thesis.

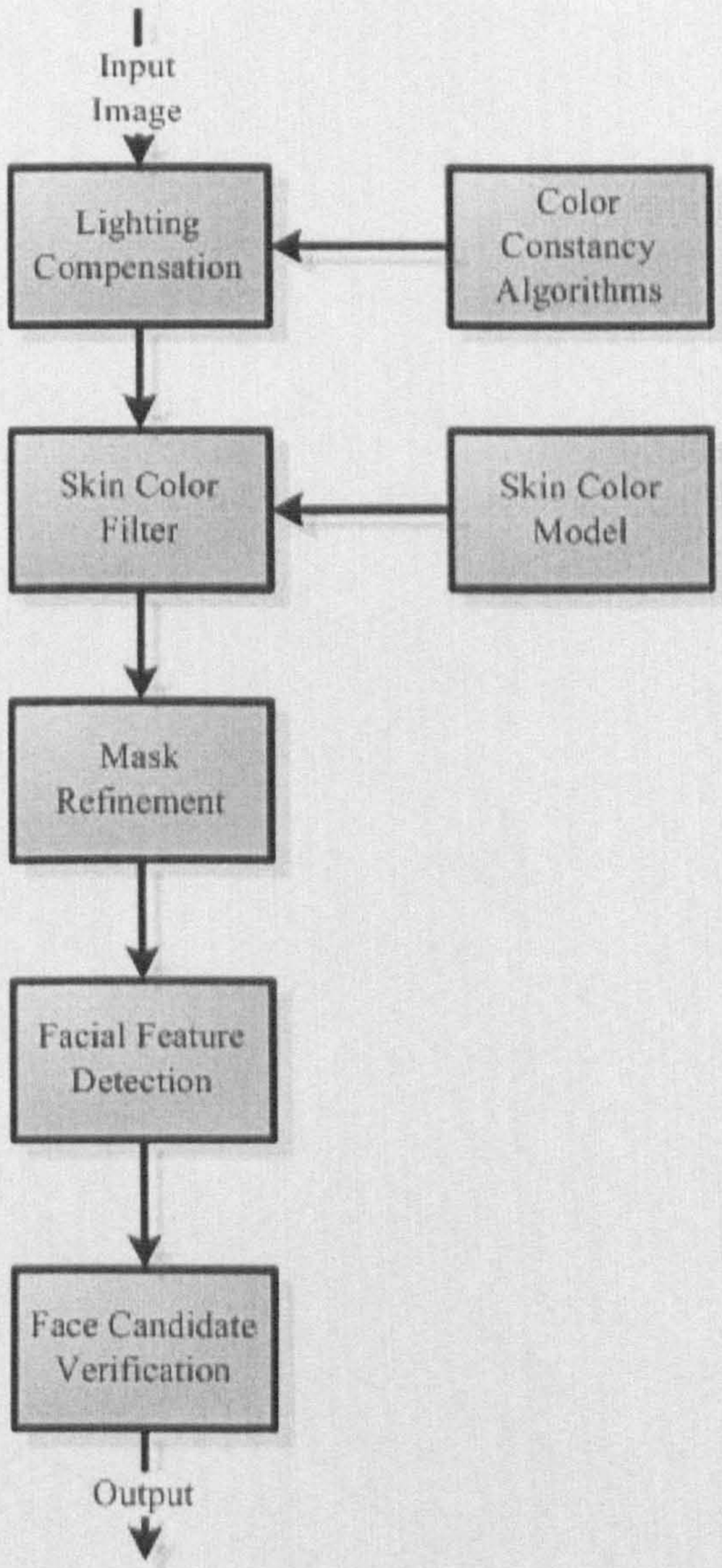



*Fig 3.1* Architecture of our face detection system

## 3.2 Introduction

Color is a low level feature that can be powerful, discriminative and computationally fast. As mentioned before, although color information has limitations and is not enough for face detection, it is still a useful cue to eliminate

obvious false objects in the background. This is particularly useful when we are going to detect faces in outdoor environments where the background would be cluttered with various objects. The challenges here include: illumination, skin color modeling and merging skin color pixels into segments.

In the upcoming sections, we would propose and analyze a more robust skin region detector that exhibits a better behavior with respect to the above problems. The goal of our skin color region detector is to achieve the highest accuracy and performance, with the least assumptions being made. The paper is organized as follows: In Section 3.3, the proposed detector is described in detail. Section 3.4 includes comparisons with other known schemes and results from the application of the proposed scheme on a variety of face databases. Conclusions are drawn in Section 3.5.

## 3.3 Proposed Scheme

Our skin color region detector has three major steps:

- Adaptive lighting compensation
- Skin color modeling and filtering
- Mask refinement

### 3.3.1 Adaptive Lighting Compensation

Due to the complex illuminated environments in the real world, a lighting compensation algorithm seems to be indispensable for a robust skin-tone color detector. With the aid of lighting compensation, the skin color distorted by illumination can be corrected to an acceptable extent. The skin-tone color boundaries in our approach can be defined more accurately as compared to other known schemes [18], [75], which will be shown in explicitly in the experimental results section.

In fact, the benefits of lighting compensation are twofold. First of all, the risk of losing a face due to illumination is dramatically lowered. This is essential since for

an automatic face detection system under unconstrained environment. Especially for surveillance systems in important facilities like airport, losing even a single face candidate is unacceptable. Secondly, with the stricter and more accurate skin-tone color boundaries, more non-face patches can be eliminated. As a result of the decreased number of false positives, the processing cost of the upcoming stages of the face detection pipeline will be decreased.

According to J.C. Terrillon et al's skin-color based segmentation and face detection analysis in different color spaces [53], the normalized *RGB* color space yield the best segmentation results and therefore result in the most robust face detection systems. Considering Terrillon's result and other comparative studies mentioned in Chapter 2, we choose the normalized *RGB* color space as the color space for our lighting compensation algorithm, as well as for our skin color model and filter.

The well known normalized *RGB* color space transformation for any channel (*R*, *G* or *B*) is shown in equation (3.1).

$$N_c = \frac{C}{R+G+B} \tag{3.1}$$

Where $N_c$ is the normalized channel value for a single channel $C\{R, G, B\}$. The triplet of the normalized channel values represents the percentage of each channel instead of its absolute value in the *RGB* color space, so it is a more constant metric with respect to minor illumination changes. From the transformation, we can discover another reason to use the *normalized RGB* color space, which is, it indeed reduces the dimensionality of the color space to 2. With $N_R+N_G+N_B=1$, the 3-D RGB color space is projected to the normalized plane, as shown in Fig 3.2. That will benefit the later process in the lighting compensation and skin color filter stages of our scheme in terms of computation reduction, since the data volume is reduced by 1/3. Other researchers like [57] generally use the normalized *R* and *G* channel, which equals a projection from the normalize plane to the *r*-*g* plane. But that would inevitably change the density of skin color region. To avoid this distortion, we choose to transform the coordinate to the normalize plane.

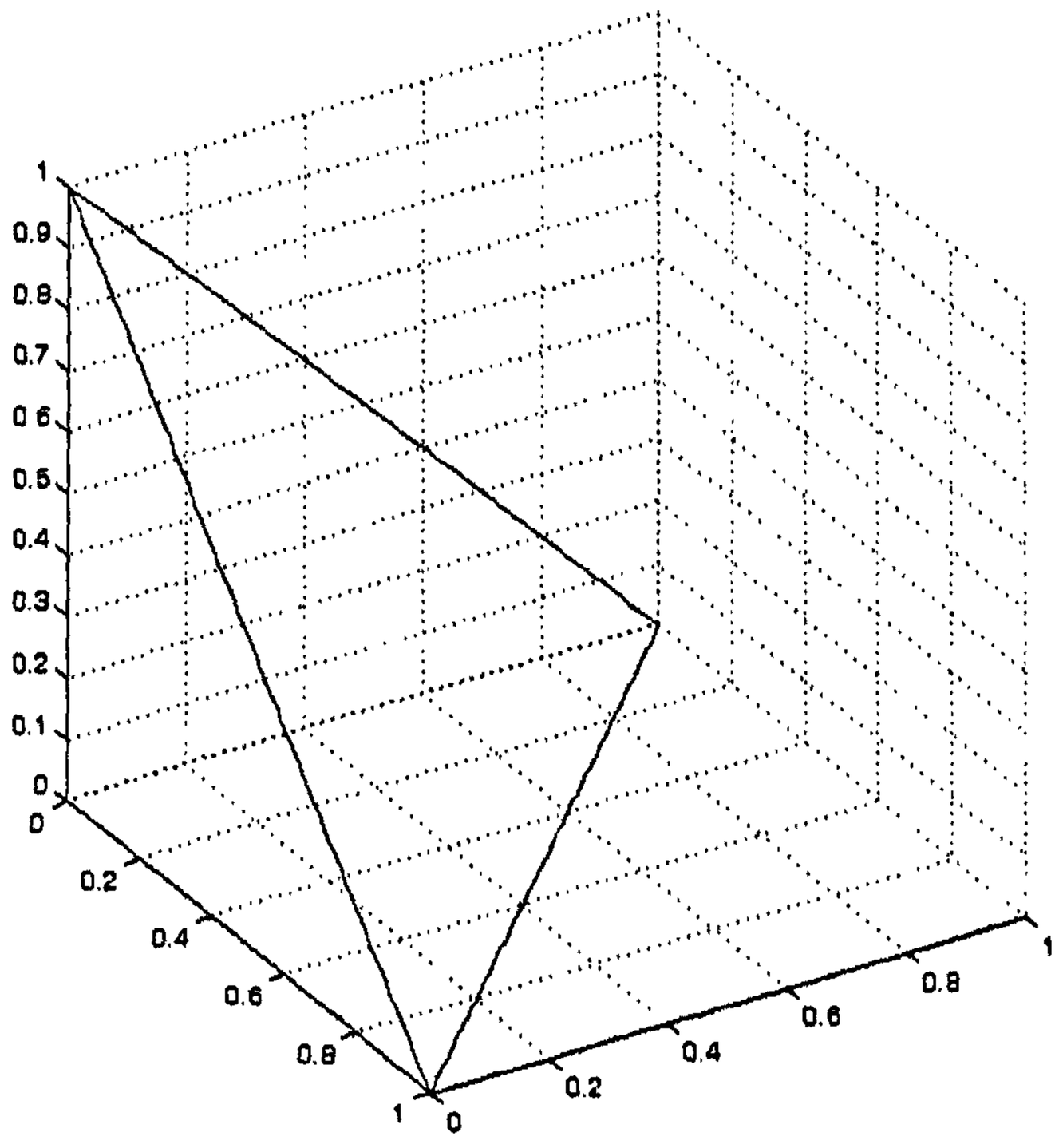



*Fig 3.2* Normalize plane

Various lighting compensation, also called color constancy algorithms, including Grey World (GW) [46], Modified Grey World (MGW) [50], White Patch Retinex [46], have been proposed to solve the problem of face detection. But most of them are defined in the *RGB* color space or color spaces that claim to be able to separate luminance form chromaticity, such as $YC_bC_r$ or *HSL*.

The Grey World algorithm is based on the assumption that the spatial average of surface reflectance in a scene is achromatic. Since the light reflected from an achromatic surface is changed equally at all wavelengths, it follows that the spatial average of the light leaving the scene will be the color of the incident illumination [54].

The MGW and White Patch Retinex algorithms are indeed modifications of the GW algorithm. The experimental results provided in [54] shows that the Grey World

(GW) color constancy algorithm performs no worse than MGW and White Patch Retinex. And without duplicate value removal and sorting operations, the GW algorithm has the lowest computation cost, so that the GW algorithm is chosen to be the basis of our adaptive lighting compensation algorithm.

The standard GW algorithm is defined as

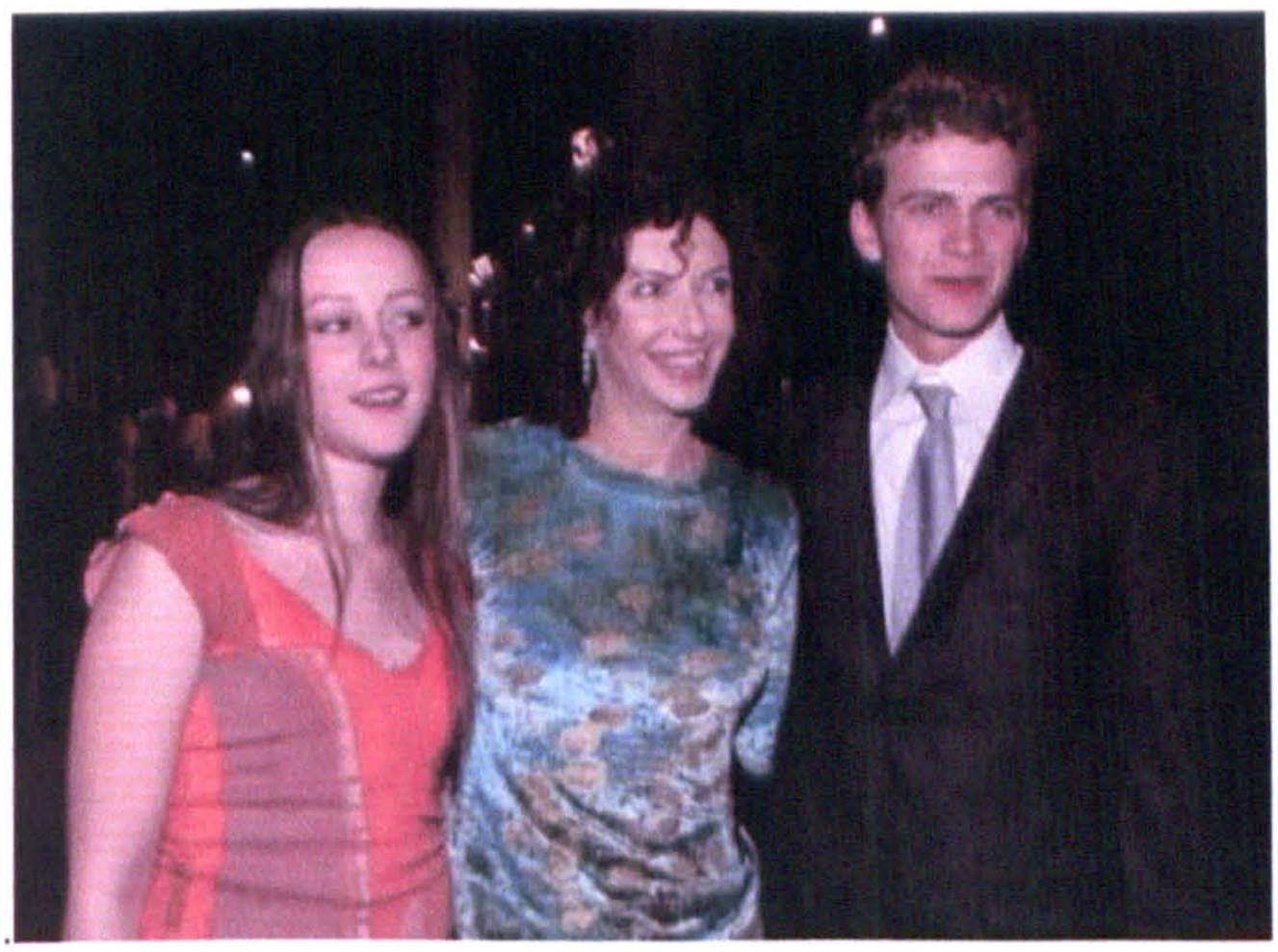
$$Sc = \frac{C_{std}}{C_{avg}} \tag{3.2}$$

Here $S_C$ is the scale factor for one specific channel $C\{R, G, B\}$, while $C_{std}$ and $C_{avg}$ are the standard mean gray value and the mean value of this channel. In spite of the variants of GW algorithms, $C_{std}$ is generally considered as a 50% ideal grey under the canonical [15]. For example, in the normalized *RGB* color space, since the maximum normalized channel value is 1, $C_{std} = 0.5$. In the usual RGB color space where the maximum value of each individual channel is 255, $C_{std}$ =0.5x255=128.

The limitation of the standard GW algorithm is obvious. From our experience, we noticed that the fixed standard mean value will not fit well for various real world images. Especially for images with dim foreground objects and a dark background, such as photos with night scenes, will be over-compensated. This problem is caused by the way that $C_{avg}$ is calculated. The mean value of each channel will be underestimated due to the black pixels, which have the value 0. And in fact, these black pixels have no contribution to reflection, so that should be considered as outliers. On the contrary, for scenes that have large bright backgrounds, in each channel the $C_{avg}$ would be overestimated. The result of the overestimation is that the brightness and contrast of the whole image will be incorrectly lowered. In essential, the central issue with the standard GW algorithm is that it will modify the image blindly, not considering if a correction process is really needed. So that when the contrast of objects in the scene is high, it tends to alter the image incorrectly.

When it comes to the normalized *RGB* color space, all the pixels with 'balanced' channels (except black ones) should have value as (1/3, 1/3, 1/3). Due to machine induced round off errors in the floating point channel magnitudes, we still consider pixels within range [0.329, 0.331] to be 'gray' due to round errors. These pixels are

already gray pixels with various intensities, and can be used to help judging if the image really needs correction. The phenomenon is shown Fig. 3.3, which has a very nice white balance. As can be seen from the histogram, each channel has most pixels fall into the range [0.3, 0.35].



(a)

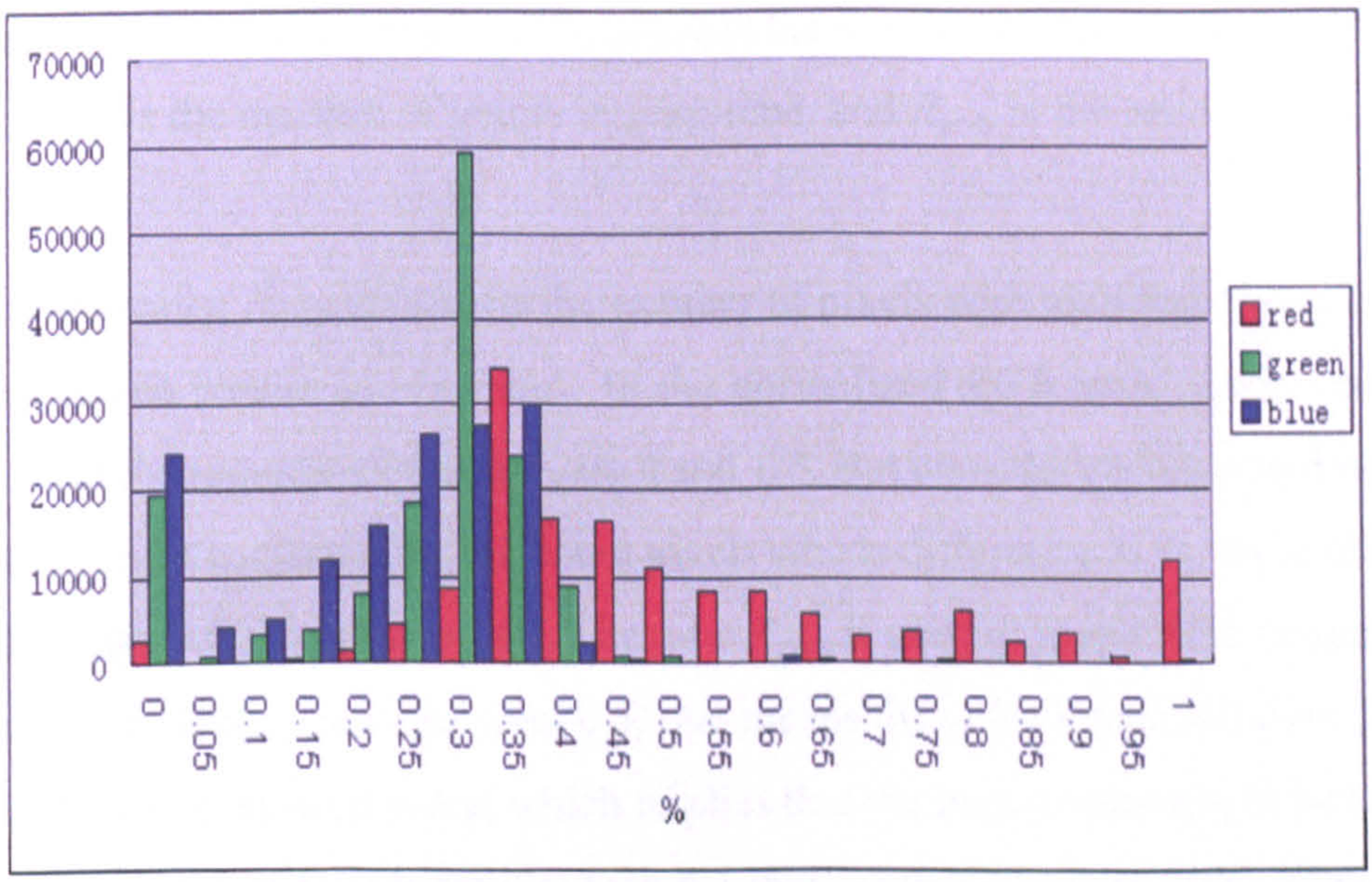



*Fig 3.3* Scene with black background (from Yahoo News photos [78]): (a) Original image (b) Normalized channel distribution

To overcome the two issues addressed above, we developed our adaptive lighting compensation algorithm. In our algorithm, the formulae we used are listed in

equation set (3.3).

$$N_{gray} = \sum_{1}^{M} \left( |N_R - N_G| + |N_R - N_B| < 0.001 \right)$$

$$R_{gray} = \frac{N_{gray}}{M}$$

$$N_{non-black} = M - \sum_{1}^{M} (N_R = N_G = N_B = 0)$$

$$C_{std} = \frac{\sum_{1}^{M} [\max(N_R, N_G, N_B) + \min(N_R, N_G, N_B)]}{2 \times N_{non-black}} \tag{3.3}$$

$$C_{avg} = \frac{\sum_{1}^{M} N_C}{N_{non-black}}$$

$$S_C = \frac{C_{std}}{C_{avg}}$$

Where $N_c$ $(C \in \{R, G, B\})$ is the normalized value of a channel, $M$ stands for the number of pixels in the image and $N_{non\text{-}black}$ is the number of non-black pixels in the image, $N_{gray}$ is the number of pixels in gray-tone, and $R_{gray}$ is the ratio of gray pixels to all pixels.

Theoretically, $N_{gray}$ should be the number of pixels with identical values in each channel, which means $N_R$=$N_G$=$N_B$. In the normalized $RGB$ color space, only two values meet the requirement, which are 0 and 1/3. But considering the rounding error in floating-point computation, we count pixels whose difference in the three channels is smaller than 0.001 as gray pixels. The ratio $R_{gray}$ is used to judge if the image needs compensation. If $R_{gray}$ is larger than 0.5, that means the color image has over 50% of the pixels are in gray-tone color, which implies that the image color might be too dull. On the other hand, if $R_{gray}$ is smaller than 0.1 implies that less than 10% of the pixels are in gray-tone color (including black and white) which is unlikely to happen. In these two cases, we consider the image needs lighting compensation, and then $S_C$ will be calculated to calibrate each channel. The value of $C_{std}$ is actually obtained from summing the dominant normalized channel with the weakest normalized channel for each pixel in the image and then dividing by two times the image dimensions. The

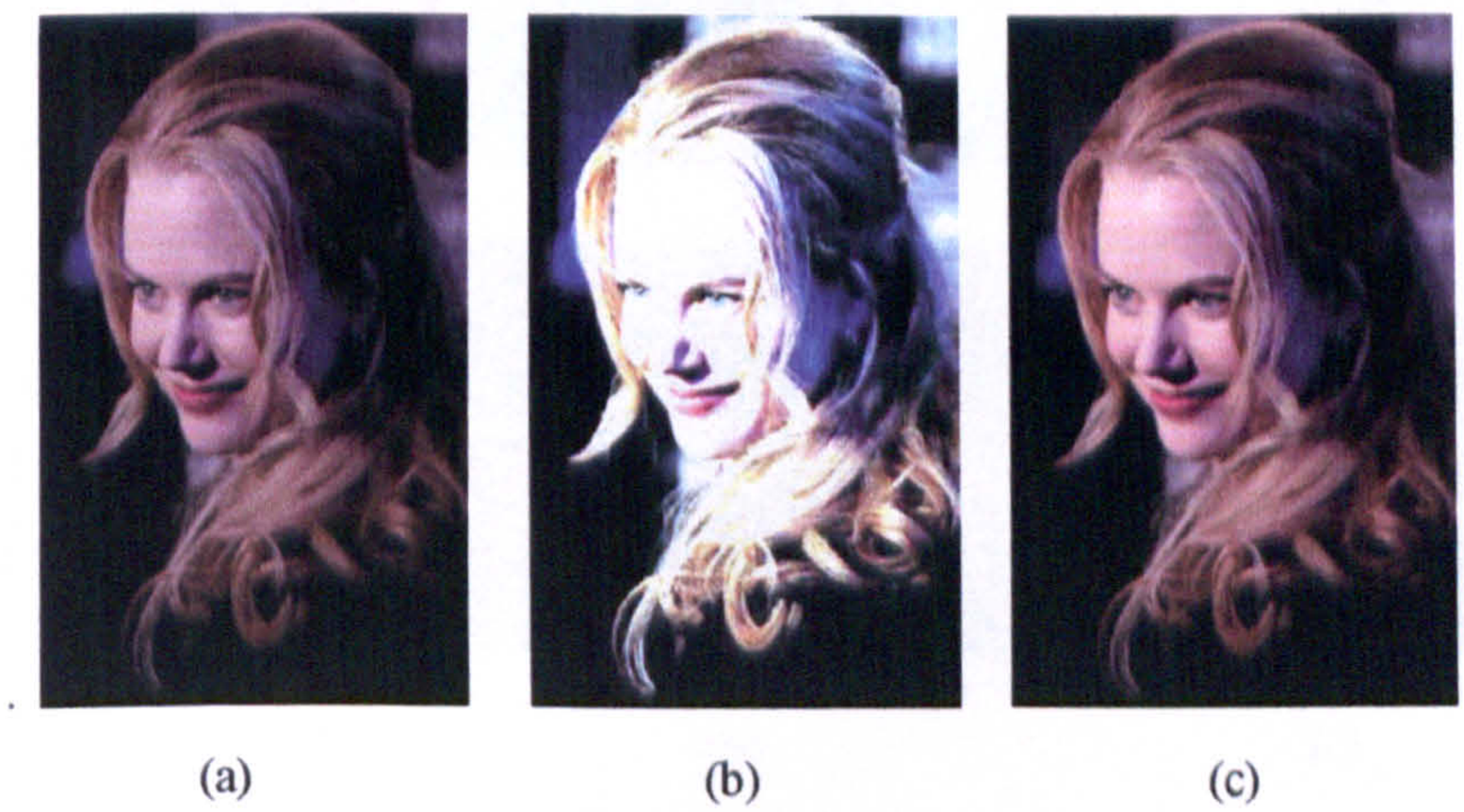
'averaging' of the dominant channel and the weakest channel is done in order to solve the over-compensation issue of the standard GW algorithm. This results in an adaptive mean gray value of the whole image. $C_{avg}$ is the mean of non-black pixels in each channel. We also limit the scale factor $S_C$ in the range [0.8, 1.2] based on empirical results.

The algorithm for light compensation can be summarized in the following steps:

a) Calculate the ratio of gray pixels in the image based on the set of equations in (3.3).

b) If the ratio is greater than 50% or less than 10%, the image needs adaptive lighting compensation. Use the set of equations in (3.3) to do this. Otherwise, lighting compensation will not be performed, which in turn saves computation. Calculate the channel scaling factors with or without light compensation.

c) Clamp the resulting scaling factors for each color channel within the range [0.8, 1.2]

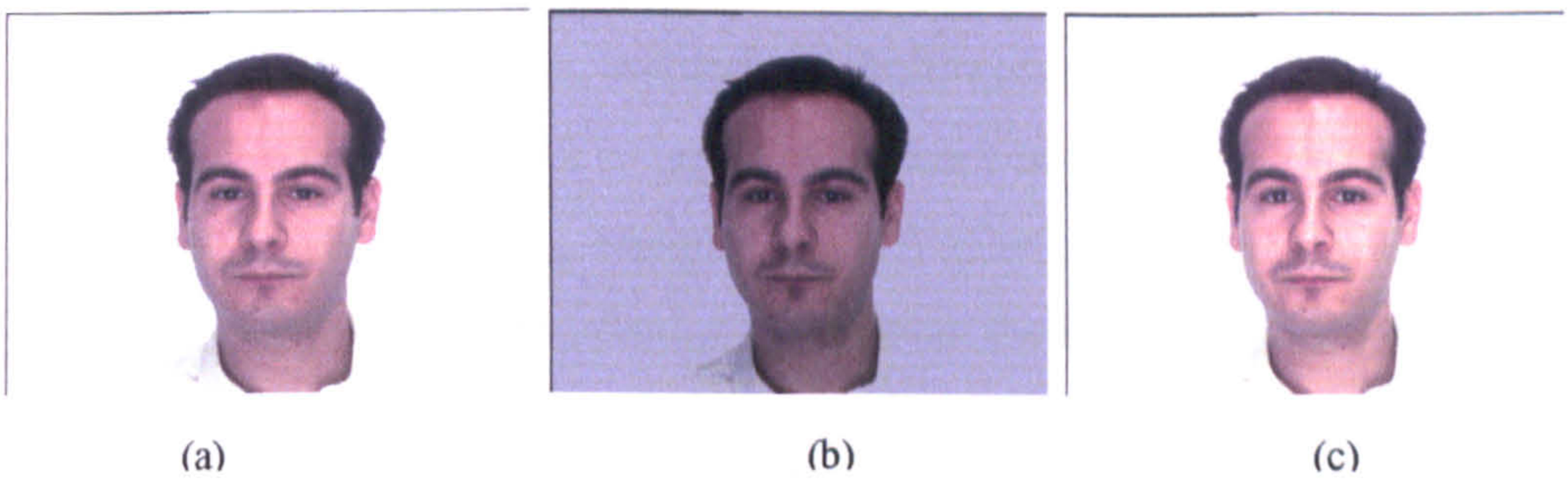


(a) (b) (c)

*Fig 3.4* Scene with black background (from Yahoo News photos [28]): (a) Original image (b) Over compensated image by standard GW algorithm (c) Result of our adaptive lighting compensation

Fig. 3.4 and Fig. 3.5 demonstrate how our adaptive lighting compensation overcomes the issues of standard GW. In Fig. 3.3(b), we can see that the standard GW algorithm over-compensates the image. It even causes chromaticity changes in

skin color and loss of facial feature details. While in 3.3(c), our lighting compensation algorithm correctly increases the contrast level on the face. When it comes to Fig. 3.4, which has a bright white background, the standard GW algorithm suppresses the value of all channels. Once again, our lighting compensation algorithm survives the challenge.



(a) (b) (c)

*Fig 3.5* Scene with white background (from Purdue AR database [29]): (a) Original image (b) Result of standard GW algorithm (c) Result of our algorithm for adaptive lighting compensation

(a) (b) (c)

*Fig 3.6* Scenes with biased color tones (from HHI MPEG7 database[26])
Upper: (a) Red-biased Image. (b) Green- biased Image. (c) Yellow- biased Image
Lower: The corresponding correction results with our adaptive lighting compensation

In Fig. 3.6, three scenes with extremely biased tone, due to suppression of color information of one or two channels, are put to test. With the help of our lighting

compensation algorithm, all the images are restored to normal tone. The results again demonstrate the robustness of our adaptive lighting compensation algorithm.

### 3.3.2 Skin Color Modeling and Filtering

Regarding the color filtering and modeling step of our scheme, the authors in [55], [56] demonstrated that the normalized *RGB* color space performs almost as well as the standard RGB space in terms of cluster compactness. This allows us to reduce the space dimensionality (from 3D to 2D) in order to achieve a simpler skin color model and reduce the computation cost in both the lighting compensation stage and the filtering stages.

To set up our skin color model, we selected around 200 faces, which include facial features and various races, from the Yahoo News database [78]. An example of cropped faces is shown in Fig. 3.7.

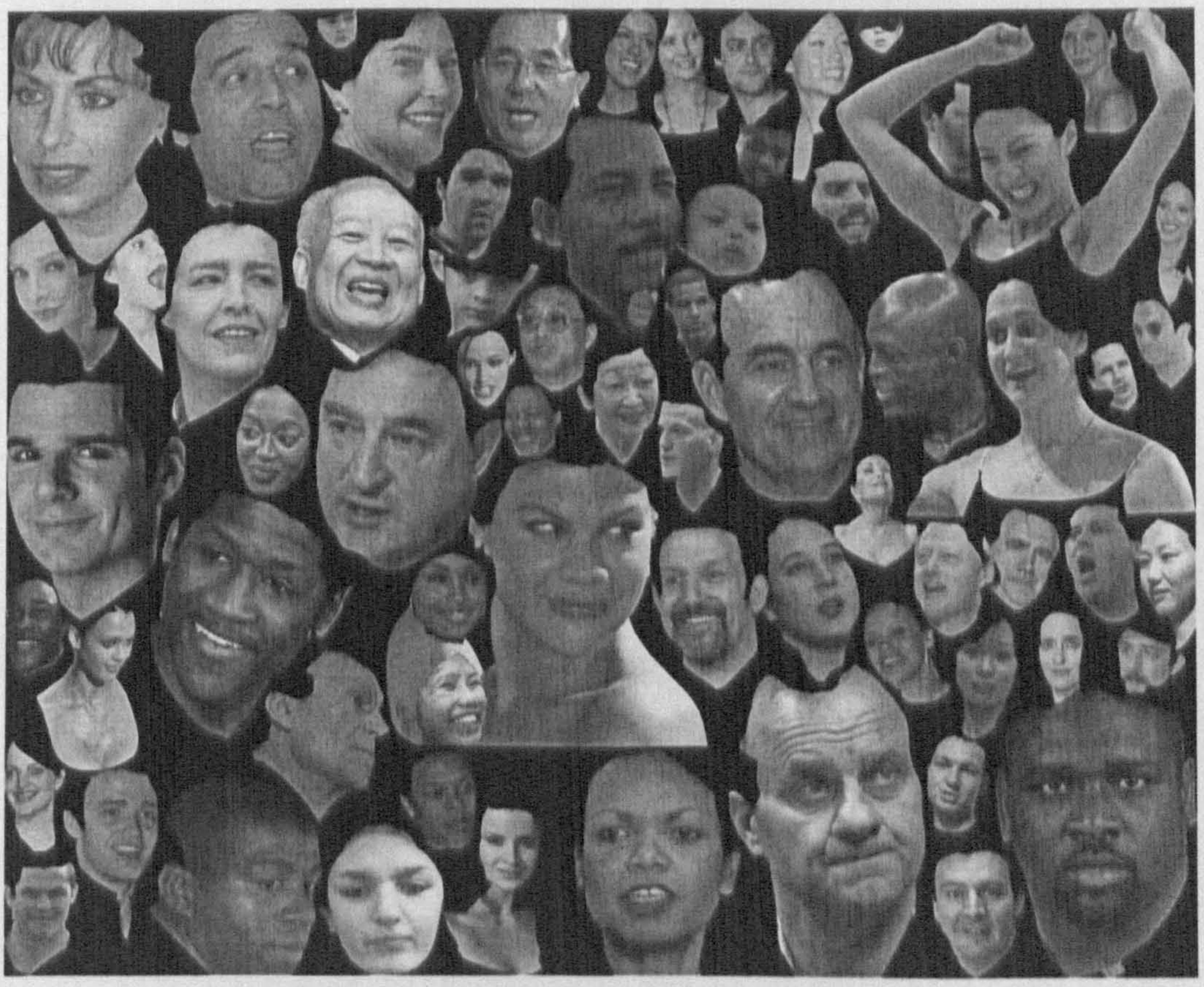


*Fig 3.7* Cropped faces to be used to setup our skin color model.

In Fig. 3.8, the skin-tone color distribution is modeled in different color spaces, which include standard *RGB*, $YC_bC_r$ and the normalized *RGB* color space we use.

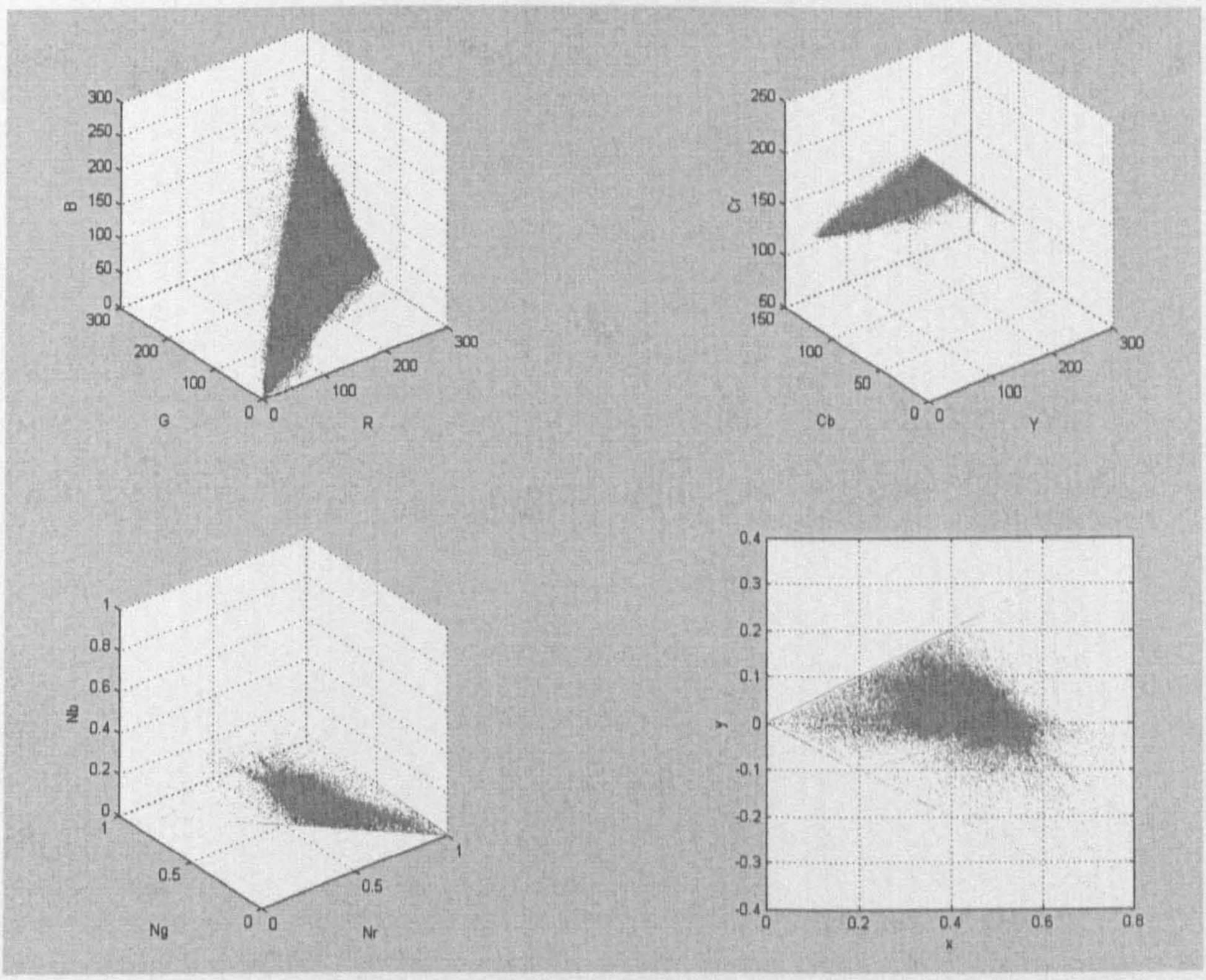



*Fig 3.8* Skin-tone color pixels modeled in different colour spaces. From top to down: *RGB*, $YC_bC_r$, normalized *RGB*, and the normalized plane we used.

As we can see from Fig. 3.7, the chosen normalized *RGB* color space has a more compact cluster of skin color than in the *RGB* and $YC_bC_r$ color spaces. Also, it is a 2-D space since all values are actually on the plane $N_R+N_G+N_B=1$. In order to avoid affine transformations and to work on the exact plane, we perform a coordinate transformation with (3.4) for each pixel. The transformed coordinates are shown in Fig. 3.8.

$$x = \frac{(1 - N_R) \times \sin\frac{\pi}{4}}{\cos\frac{\pi}{6}}$$
$$y = \frac{(N_G - N_B) \times \cos\frac{\pi}{4}}{\cos\frac{\pi}{6}} \tag{3.4}$$

After the transformation, the normalized $R$ channel then becomes the x axis, with reverse direction. So that the higher the normalized value of the $N_R$ channel, the closer it is to the origin of the new coordinate system. The $y$ axis of the new coordinates is decided by both the normalized $G$ and $B$ channels as shown in the equation above. Considering the transformation formulae for $y$, the numerator consists ($N_G$-$N_B$). So if the magnitudes of the two channels are 'balanced', the $y$ value would tend to 0. The reason to transform the new coordinates system in this way is as follows: It is well known that the skin color would tend to red even when there are ethnic differences, because there is blood running under skin cells. This implies that red would be the dominant color component.

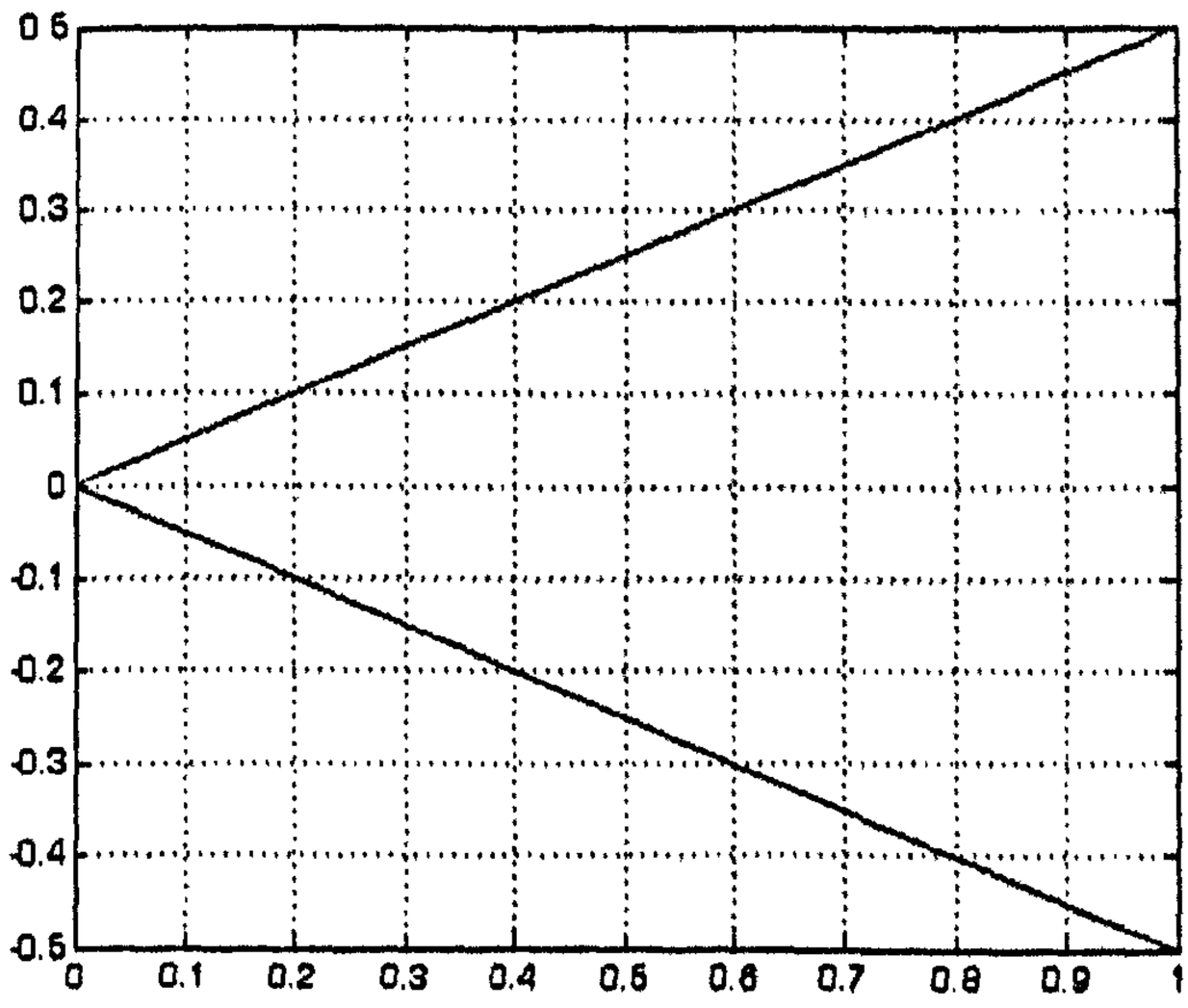



*Fig. 3.9* The coordinates used.

After the coordinate transformation, we have the skin-tone color distribution on the plane $N_R+N_G+N_B=1$, as shown in Fig. 3.10. And from Fig. 3.10(a) we can see

that, there are actually many outliers in our model. They are mostly caused by the facial features including facial hair, eyes, lips, and etc. To remove the outliers, we apply a morphological opening filter with a diamond-shape structure element, considering the skin color distribution as an image. From (c) and (d), we can see there are some obvious vertexes in the model.

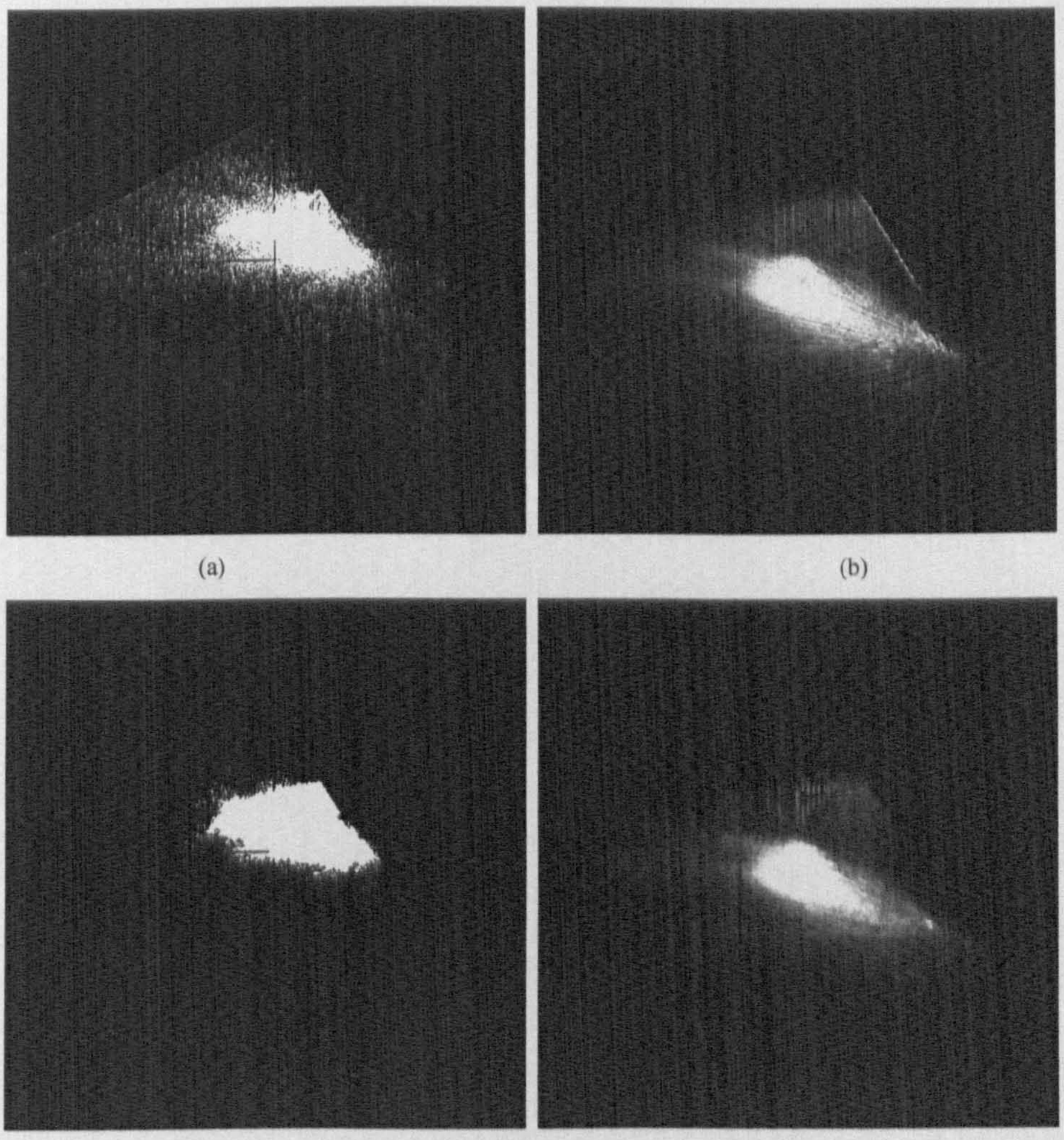



*Fig. 3.10* (a) The binary image of the modeled skin color (b) The density map of the model (c) The binary image of the model after applying the filter (d) The density map of the filtered model

These are the boundaries of the skin color cluster we use for our skin color filter. The skip pixels are bounded by a pentagon with vertices (0.31, 0.05), (0.45, 0.1), (0.542, 0), (0.5, -0.2) and (0.38, 0) as shown in Fig. 3.11. And the equations are listed in (3.5):

$$\begin{cases} y \le -0.7134x+0.2714 \\ y \le 0.3571x-0.0036 \\ y \le 1.6667x-0.6333 \\ y \ge -0.2065x+0.0511 \\ y \ge 4.7619x-2.5810 \end{cases} \tag{3.5}$$

Some results of our skin color filter on different skin-tone colors are shown in Fig. 3.12. Our skin color filter can well adapt to different skin-tones of different races. Another example is given in Fig. 3.13 for more white and black athletes. Our skin color filter works well with the upcoming mask refinement step.

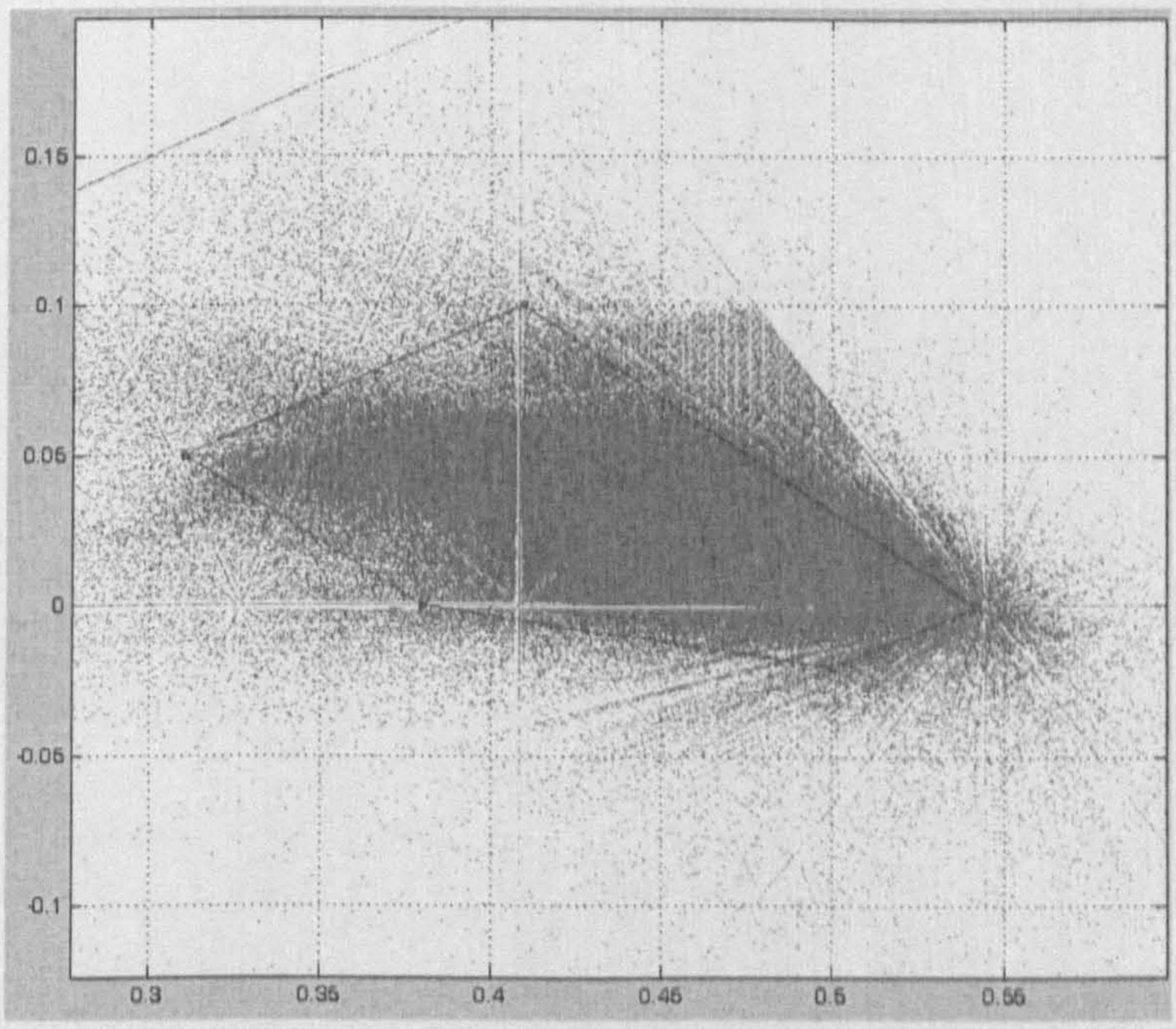


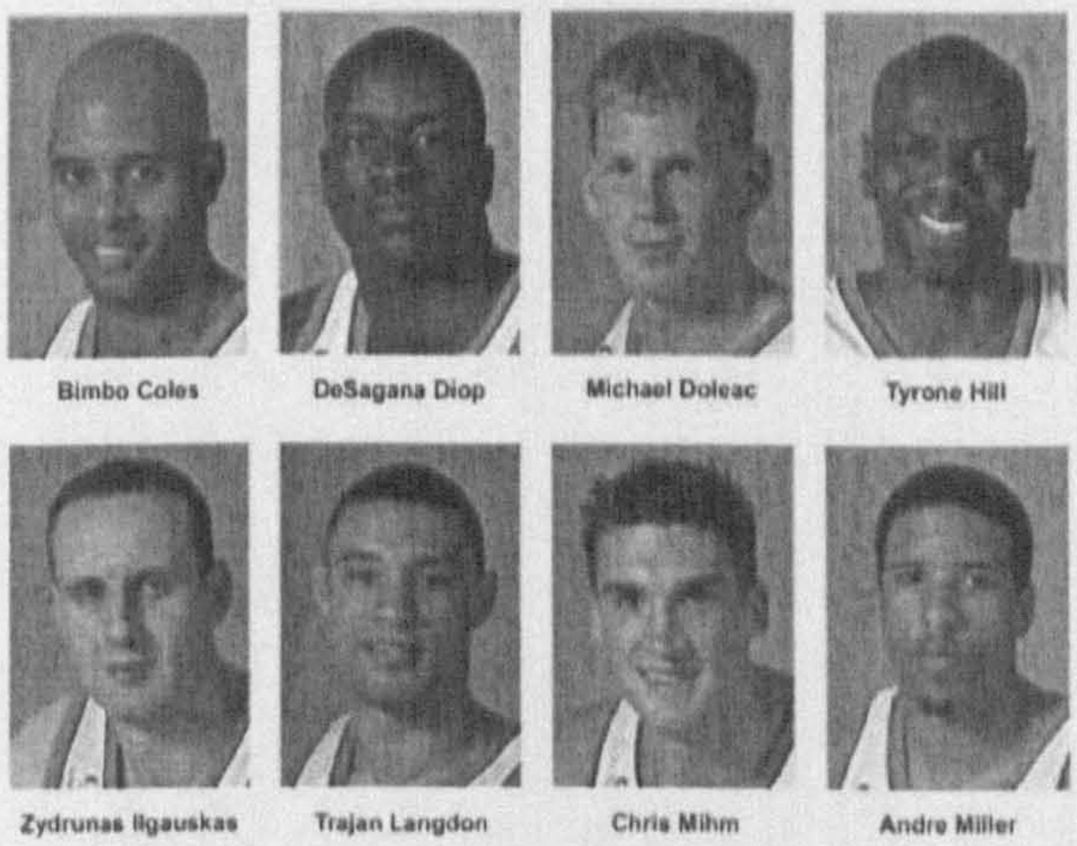

*Fig. 3.11* Boundary gained from skin color filtering.

(a) (b) (c)

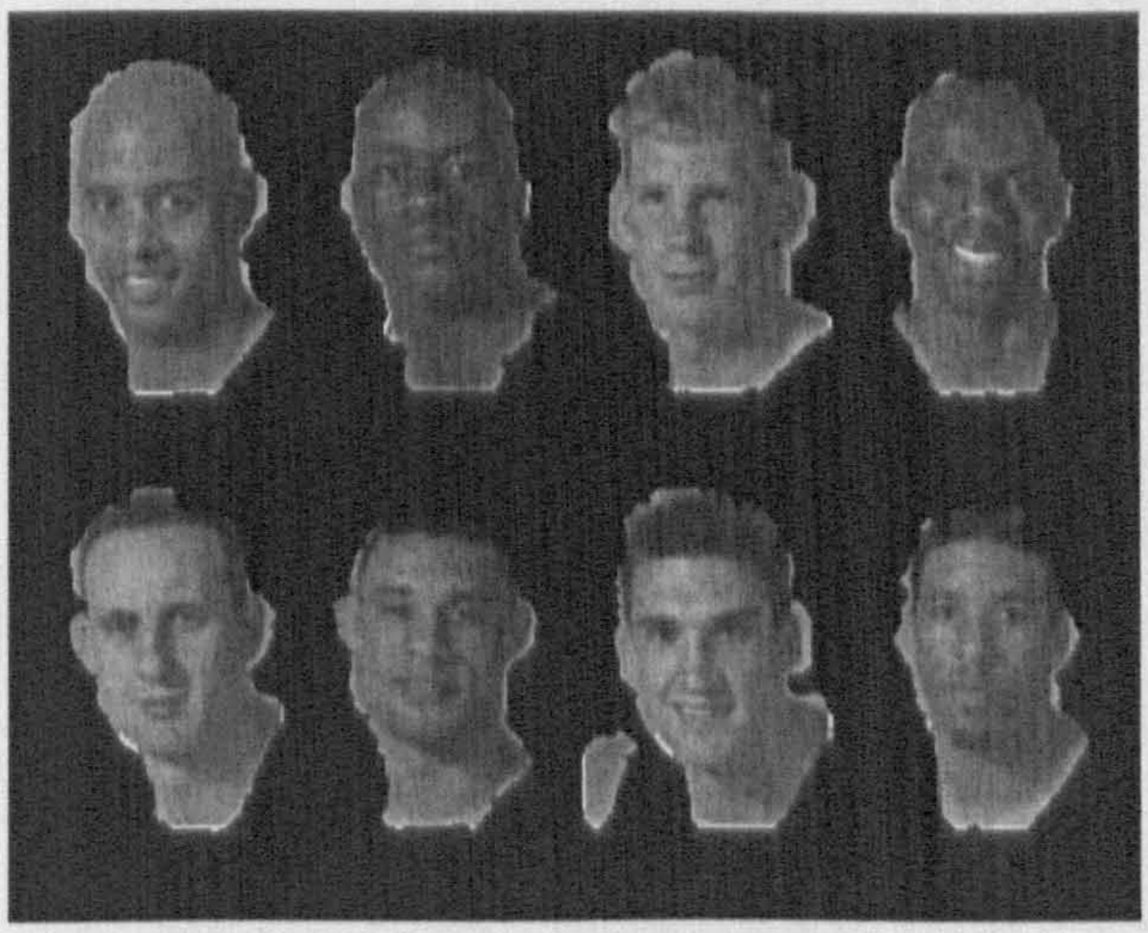
*Fig. 3.12* Skin color filter result for different skin colors (a) White (b) Black (c) Yellow
Upper: Original images
Lower: Skin filter + flood filling results

We understand that skin color modeling plays a very important role in our face detection system architecture, and there are other maybe better modeling approaches available. Given that our architecture is an open one, for a real system, we can choose the skin locus to achieve a device-optimized skin color model which would be more appropriate for the application.

The restrictions of our skin color filter are loose since this would give us an advantage in terms of avoiding the loss of face candidates. As a tradeoff, this brings a dramatic increase in the number of false positives. To eliminate these false positives, some post-processing steps are needed.




(a)



(b)

*Fig. 3.13* Our skin filter applied to athletes. (a) Original image (b) Skin filter + flood filling result

### 3.3.3 Mask Refinement

Regarding the mask refinement stage of our scheme, existing skin-tone color filter algorithms like [18], [23], [44], [45], have no further processing of the produced mask after the skin color filtering. In consequence, lots of noise pixels which do not belong to the skin regions remain. The retention of these irrelevant pixels will in turn place heavy computational loads on subsequent stages of the face recognition process, such as the feature extraction and face verification.

We believe that in the real world, where images have complex backgrounds and

illuminations, it is unrealistic to expect that the skin color filter itself, no matter how sophisticated it is designed, can eliminate all the non-skin pixels that lie in the background and other objects. In this sense, further processing is needed. Towards a solution to the noise issue, or at least to suppress it, we choose to utilize morphological operators [58], [69].

The assumption for our mask refinement scheme is simple. We assume that contours of human body parts, especially faces, to be smooth curves. That means any jagged contour; sharp corners or protrusions would be smoothed. Also, there should not be holes in the face skin area that do not belong to the face itself. We consider our assumption a common knowledge of human faces.

The foundation of morphological image processing is the Mathematical Morphology theory, which is built upon Lattice theory and topology. Morphological operators are image operators that based on shift-invariant (a.k.a. translation invariant) operators, principally on Minkowski addition, named after Hermann Minkowski. Minkowski's addition of two sets X and Y is defined as:

$$A \oplus B = \{a + b : a \in A \text{ and } b \in B\} \tag{3.6}$$

Fix a point $O$ in the plane. Point $O$ is called the origin. The directed segment $OA$ from the origin to an arbitrary point $A$ in the plane is known as the $A$'s radius-vector. Radius-vectors of two points can be added according to the rule of parallelogram. To find Minkowski's sum of two sets one must consider the totality of all possible sums of a point from one set and a point from the other. If the origin is translated from point $O$ to $O'$, the sum of two sets is translated by the same distance, but in the opposite direction. A visual representation of the Minkowski addition is shown in Fig. 3.14.

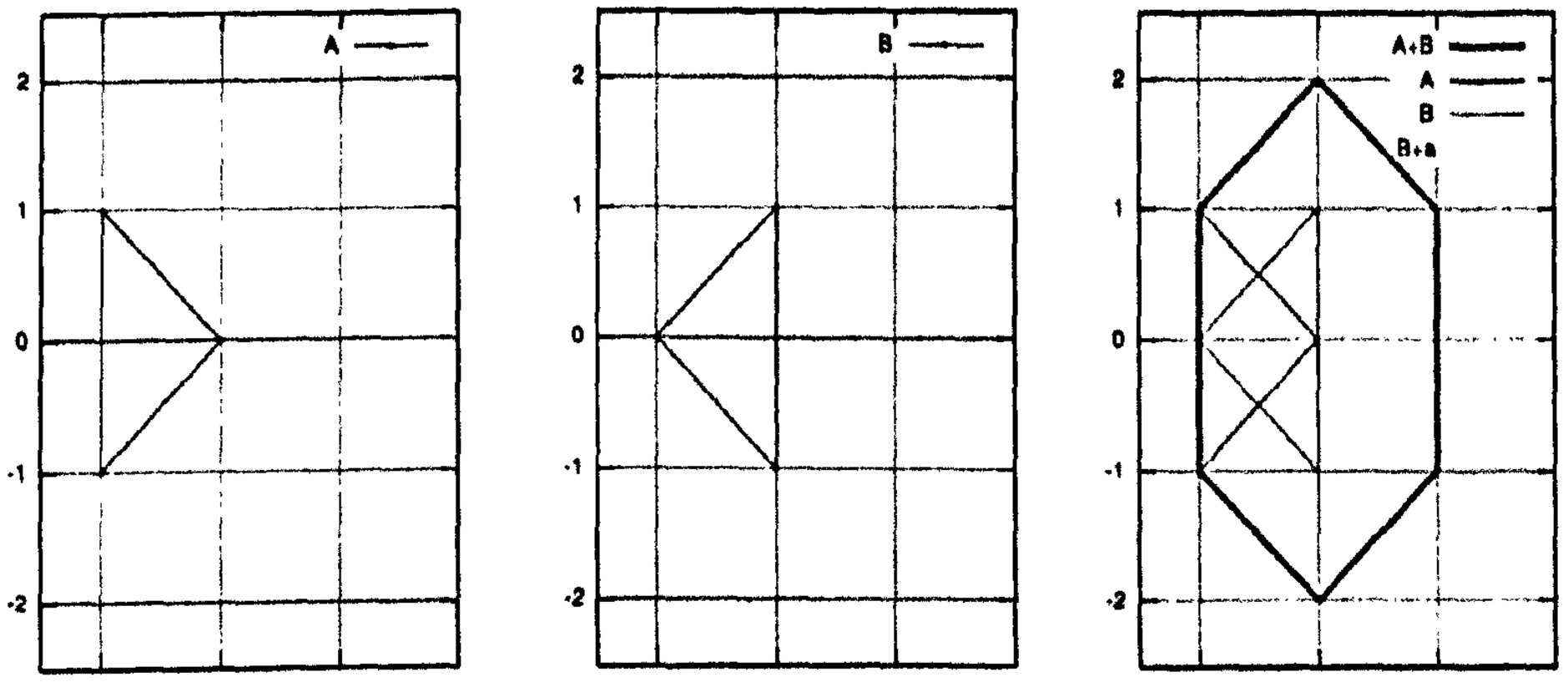



*Fig. 3.14* Minkowski addition. (a) shape A (b) shape B (c) Minkowski addition of A and B

Minkowski addition is also called the binary *Dilation* of A by B. In contrast, the binary *Erosion* can also be defined using Minkowski addition as:

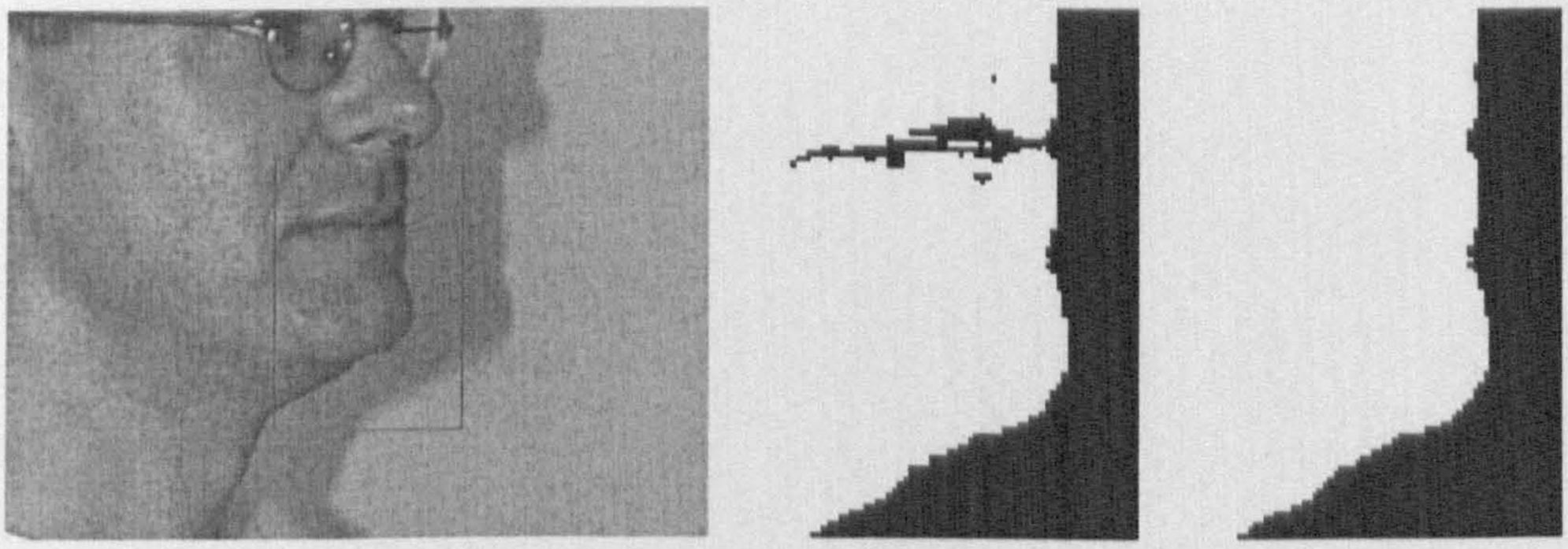
$$A - B = (A^C \oplus B)^C \tag{3.7}$$

In practice, the original image is considered as $A$, and $B$ would be the structure element used to smooth the shape. To solve the problem when the structure element reaches the edge of the image, a padding scheme is applied. For the dilation operator, pixels beyond the image border are assigned the minimum value afforded by the data type. For binary images, these pixels are assumed to be set to 0. And for the erosion operator, pixels beyond the image border are assigned the maximum value afforded by the data type. For binary images, these pixels are assumed to be set to 1.

The particular sequence of our morphological operators is as follows. First, a dilation operation is applied, with a diamond-shaped structure element. Small gaps would be reconnected with this 4-connected structure element. This operation would result in the expansion of contour patterns. In some cases, the contours of the patches may contain important facial features such as eyes or lips on the boundaries of the patches which have been potentially filtered out by the skin color filter, due to their

different color. This situation is mostly dominant in profile style facial images.

The second step is the flood fill operator. 8-connected contours with a hole in it will be filled by this operator. From Fig. 3.15(b) and 3.15(c), we can see how the dilation operator manages to reconnect the separated contour of the skin patch due to lip color. With the reconnected contour, the lip area would be regained by the later flood fill operator.



(a) (b) (c)

*Fig. 3.15* Facial features on the contour. (Image from HHI MPEG7 database [26]) (a) Original image. (b) Result after skin color filter (c) Result after mask refinement

The third step in our scheme is the opening operation. The property of this operator is that it is anti-extensive and will not introduce new edges in the contours. Its main role is suppressing sharp protrusions and the eliminating narrow passages.

For both the dilation and opening operators, we choose diamond-shape structure elements, which are close to disc-shape structuring elements in discrete spaces (see Fig. 3.16). Also, the disc-shaped structure element will recess to diamond-shaped ones in small granularity. An important property of the diamond-shaped structure elements is that they make the morphological operations orientation invariant. One more reason to use the diamond-shape structure element is that it can be decomposed in to straight line structure elements [69]. The logarithmic decomposition can be used to significantly accelerate the morphological transformation process.

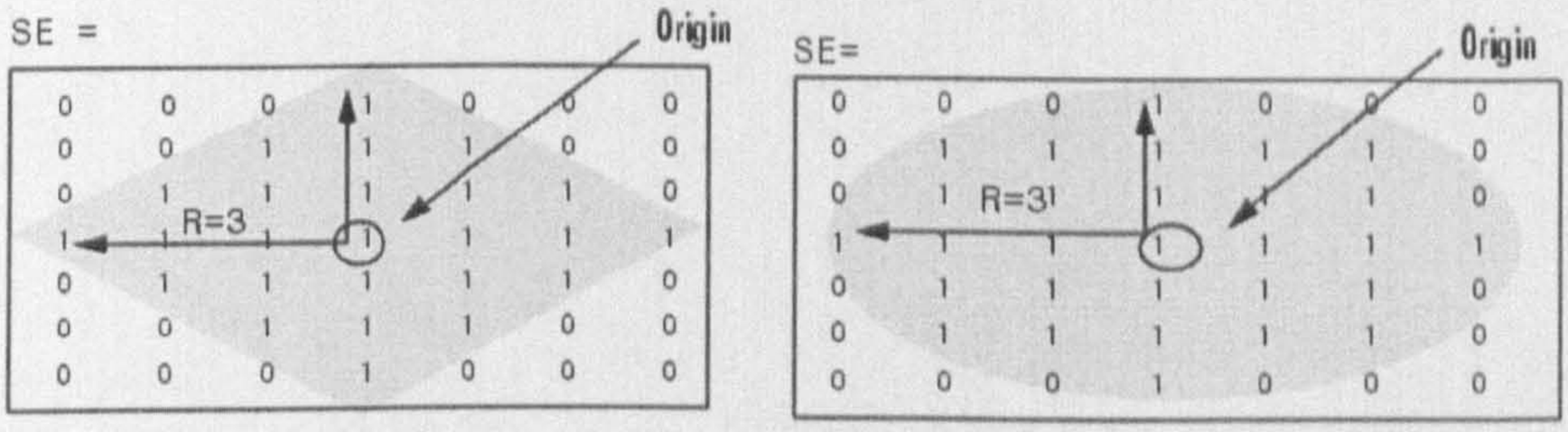


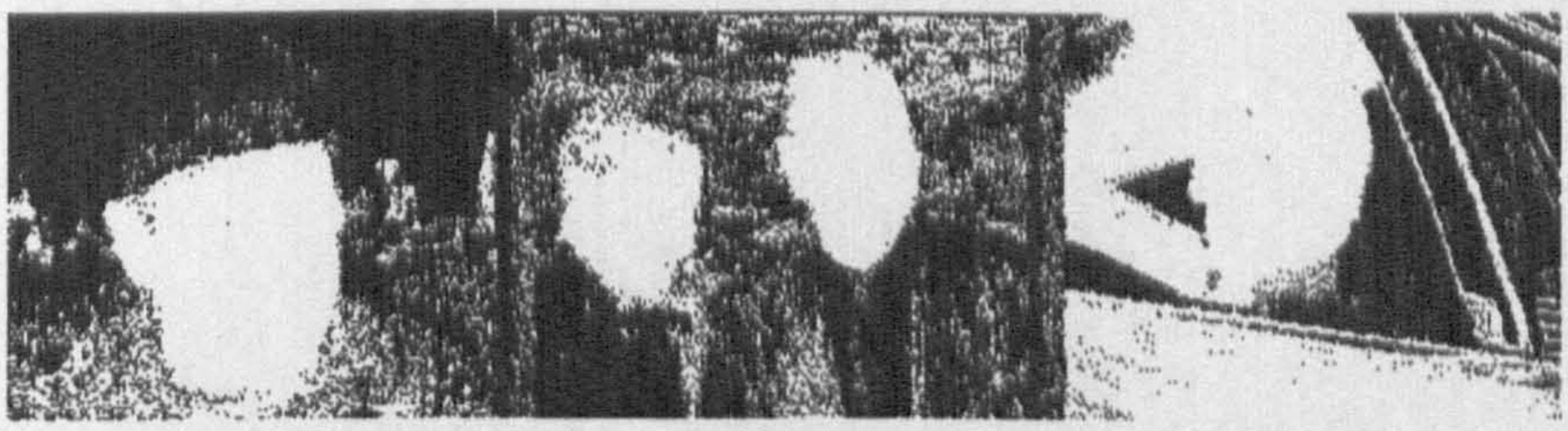

*Fig. 3.16.* Structure elements (images taken from MATLAB help document) (a)diamond shape (b) disc shape

Finally a binary area opening operator is used. For binary area opening, we consider objects that have less than 120 pixels as noises. From [7] we know that nowadays, the smallest size of face on which existing face detection algorithms are applied is 11 by 11, which means at least 121 pixels are needed for a face candidate to be verified as face. On the other hand, if the resolution of the face is too low, even if it can be verified, it cannot provide enough facial feature information for recognition use.

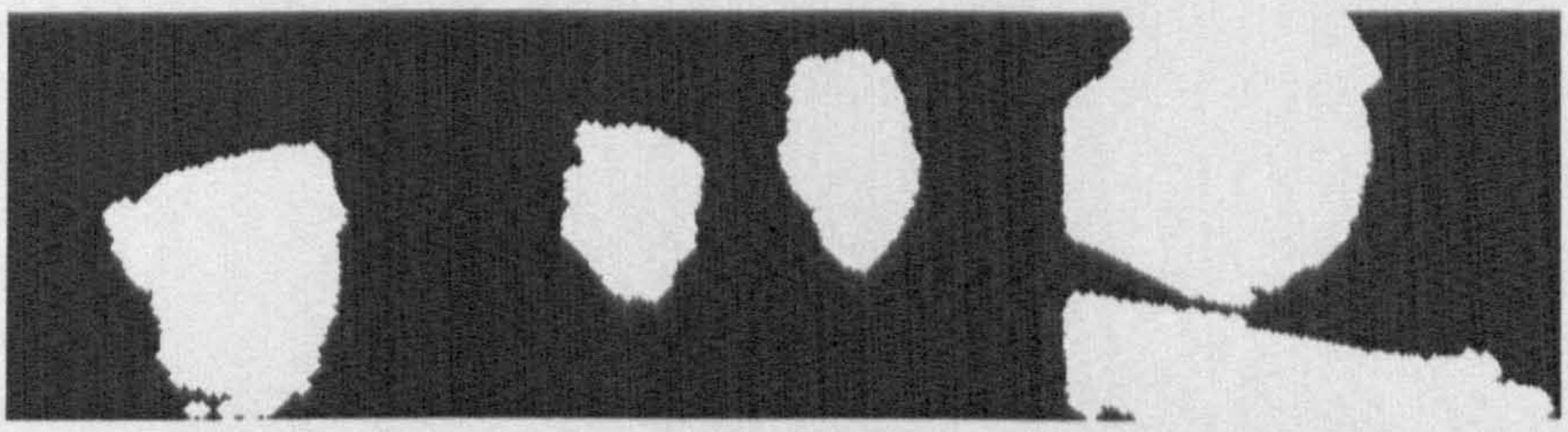
By using a carefully chosen combination of morphological operators, our scheme gains several advantages:

Firstly, most of the noises in the background can be successfully removed. This would benefit our skin color filter design since we can loosen the restrictions of the skin color boundary definition with re-application of skin tone modeling and filtering, thus leading to higher detection rates. As it is well known, there is always a tradeoff between false negatives and false positives when designing a skin color filter. In our scheme, we can boost the detection rate by leaving the false positives of the modeling and filtering stage to be suppressed in this stage. From Fig. 3.17, we can see that our mask refinement scheme is effective since sparse dot shaped noises, together with the line shape noises in the background, i.e. the fringe of the car window in the third image, are completely removed.

(a)



(b)



(c)

(d)

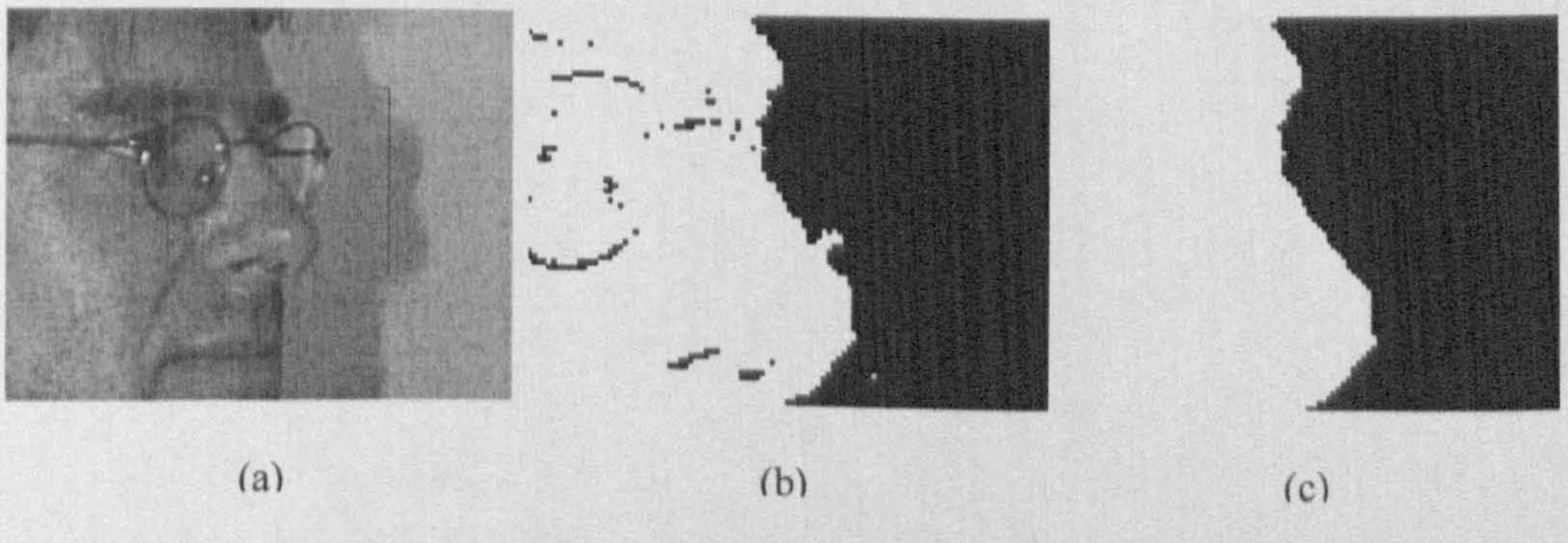
*Fig. 3.17* Background noises removed by mask refinement (a) Original image. (b) Mask gained after applying skin-tone colour filter. (c) Mask gained after mask refinement. (d) Final result

Secondly, the mask refinement process also enables the retention of important facial features such as eyes, nostrils, lips, and also bright/shadow spots on the face, which obviously have a different color to the ordinary skin. With the help of the flood fill operator, holes in the mask would be filled, so that the aforementioned features would be retained. An example for the facial feature retention is given in Fig. 3.18.

(a) (b) (c)

*Fig. 3.18* Facial fetures retained by flood-fill operator (a) Original image. (b) Result of Sandeep et al[ [71] without filling (c) Result after mask refinement

Last but not least, the mask refinement stage can help to overcome the artifacts issue in JPEG format images. Nowadays JPEG is the most popular image compression format, but unfortunately, it is block based. That will inevitably cause some artifacts on the edge of the segment gained by skin color filter. The mask refinement can remove these artifacts by smoothing the edge of the segment, and in turn achieve a better contour segmentation. An example is given in Fig. 3.19. In Fig. 3.19, we can notice that a part of the glasses frame remains after skin color filter. It should have different color to skin, but because it is in the same block with adjacent skin pixels, it is coded as the skin color.



(a) (b) (c)

*Fig. 3.19* JPEG artifacts and removal (from HHI MPEG7 database [26]) (a) Original image. (b) Result after skin color filter (c) Result after mask refinement

The comparison of our scheme to other published works, like [18], [45], will be

shown and analyzed in the experimental results section.

## 3.4 Experimental Results and Analysis

### 3.4.1 Performance Analysis

Compared to the scheme proposed by Garcia et al [45], our scheme has much lower algorithmic complexity as shown below.

For an arbitrary image consisting of n pixels, the complexity of our algorithm is:

$$O_p(n) + O_m(n) + O_f(n) \tag{3.8}$$

While the complexity of Garcia's algorithm is:

$$O_s + O_q(n) + O_l(n) + O_p(n) + O_f(n) \tag{3.9}$$

Where $O_s$, $O_q$, $O_l$, $O_p$, $O_f$, and $O_m$ respectively represent the operations for color table setup, color vector quantization, skin color table lookup, pixel evaluation, pixel filtering, and morphological filtering. It is well known that the clustering technique in [45] entails a color table setup which is computationally expensive. Also it is not guaranteed to produce accurate results, since it depends on the different data samples used and on the choice of various initial points for clustering [72]. In implementation terms, table lookup operations will cause lots of I/O operations, and vector quantization operations are also computationally expensive.

In the algorithm we propose, the mask is a binary image. In most cases, it can be stored in memory and morphological operations performed on it will be extremely fast. It has to be noted that even if we ignore the operations regarding the color table setup procedure in Garcia's algorithm, still $O_m << O_q + O_l$.

Furthermore, our scheme can achieve further speed-ups by applying the new queue-based contour processing algorithm, as presented in [68], which is based on fast binary neighborhood operations for accelerating morphological processing. The logarithmic decomposition of the diamond-shaped structure element can bring it further and make parallel processing possible.

### 3.4.2 Experimental Results and Comparison

In the skin color modeling section, we obtain the skin color region for our model from 200 faces examples. We choose to enlarge and shrink the skin region area by moving the vertexes locations. In the case of enlargement, the two vertexes on the $x$ axis move out of the pentagon for value 0.05 horizontally. And the other three vertexes move vertically out of the pentagon for value 0.05. For the shrinking case, they move in the opposite direction with the same step length. The comparison of the region change is listed in Table 3.1.

TABLE 3.1
COMPARISON OF CHANGING SKIN COLOR REGION ON CHAMPION DATABASE [77]

| Scheme | Current | Enlarge | Shrink |
|---|---|---|---|
| Stage 1. Skin Color Filter | | | |
| No. of FP | 69,551 | 92,139 | 110,269 |
| DR (%) | 99.53% | 100% | 67.74% |
| Stage 2. After Processing | | | |
| No. of FP | 1,801 | 2,471 | 918 |
| DR (%) | 99.53% | 100% | 67.74% |
| Final FP avg. | 1.421 | 1.952 | 0.725 |

FP: False Positive, DR: Detection Rate

From Table 3.1, we can find out that enlarging the skin color region increases the detection rate. Yet at the same time, it introduces a lot of false positives, which is 37% more than the current one. On the other hand, shrinking the skin region reduces the false positives, but the detection rate fall dramatically to 67.74%, which is far below the acceptable level. From the data we can know, the current skin color model is rather 'optimal' for general use.

In [75], Chai and Ngan proposed a color-based scheme specifically designed for face segmentation in videophone applications. It works on fixed CIF-sized images and is not a face detection system since they assume there must be faces in the

images. In this sense, our comparison with that scheme is only in terms of face segmentation accuracy.

The major stages of Chai's scheme includes: color segmentation, density regularization, luminance regularization, geometric correction and contour extraction. After the skin color filtering, density regularization is applied. 4x4 blocks are used to build the density map and the density value is quantized from 0 to 16. Then, if a full-density block can remain in the extracted segment depends on its neighbors. This is somehow similar to the morphological operators we used. But without the lighting compensation, the skin color filter and density map cannot guarantee the right segmentation of face from the background. As illustrated in Fig. 3.20, without the lighting compensation, it is hard to achieve the correct face segmentation.

(a) (b)

*Fig. 3.20* Image and mask gained after skin-tone color filter that the density map approach will fail. (Image from HHI MPEG7 database [76]) (a) Original yellow-biased image. (b) Mask generated by skin color filter without lighting compensation

In Chai's scheme, for the geometric correction stage, any patch with less than four horizontally or vertically connected blocks will be eliminated. This is indeed assuming a face has the minimum size 16x16, which means, it is scale dependent. Also, the horizontal and vertical scan assumes the face is upright. That implies that the scheme would also be rotation and orientation dependent. In our scheme, the morphological operators we used have the scale, rotation, and orientation invariant properties, which bring more robustness in our system.

Fig 3.21 compares our scheme with Chai's method. As can be seen, our scheme can correctly segment the face from the background, without the verification process based on more assumptions. So far, if our skin region detector is used to alter Chai's algorithm, we can already have similar results.

(a) (b) (c)

*Fig. 3.21* Segmentation accuracy comparison with the Ngan's approach (a) Input image (b) Mask produced by Chai's scheme (c) Mask produced by our scheme

Hsu et al's work [18] has a similar architecture to our scheme. In their work, the face localization section includes five steps: lighting compensation, color space transformation, skin color detection, variance-based segmentation and connected components grouping. This is considered as stage one. After the face localization, rectangle merging is applied to obtain the face candidates for feature extraction, which is considered as stage two. In our scheme, there are three steps. To have a head-on comparison with Hsu's two-stage architecture, we group our lighting compensation and skin color filter as stage one and we name it skin color filter. The mask refinement belongs to stage two which is named after processing stage.

First of all, the lighting compensation method introduced by Hsu et al in [18] uses "reference white" to normalize the color. This originates from the White Patch Retinex algorithm. There are two assumptions here:

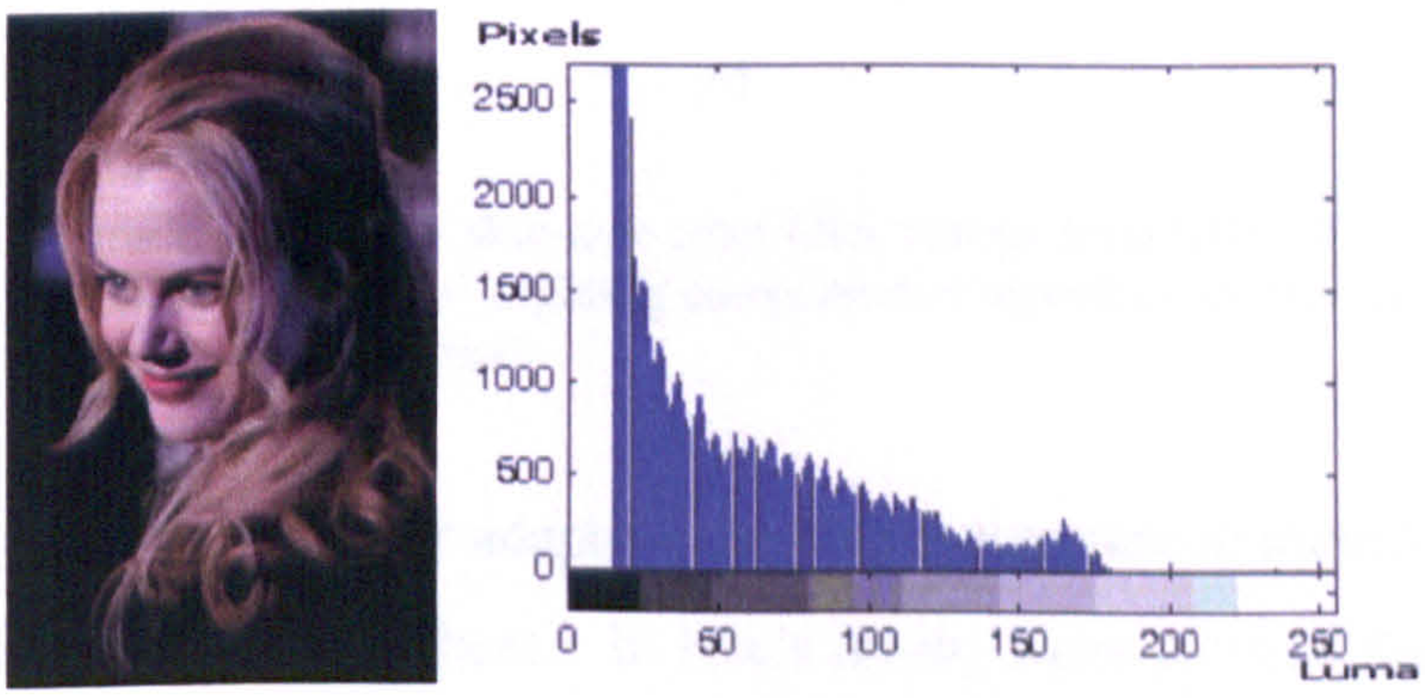
1) An image usually contains "real white".

2) The dominant bias color always appears as "real white".

Hsu considers the pixels with top 5% of luma (nonlinear gamma-corrected luminance) values in the image as the reference white only if the number of those

pixels is sufficiently large (>100). It is also claimed that the image will not be changed if the sufficient number of reference white pixels are not detected or the average color is similar to skin tone.

The lighting compensation works fine in Hsu's work, but it is not as robust as the adaptive lighting compensation used in our scheme. In our opinion, the first assumption they used is general. But the second one is stricter than the achromatic assumption of our scheme. For the top 5% luma as reference white criteria, we can find some exceptions. The exceptions would be scenes taken in night and the foreground objects are underexposure. Here we take the image we used before in Fig. 1 as an example. As shown in Fig. 3.22, none of the pixels falls in the top 5% luma value (range from 200 to 250 on the X axis). So the fact that this image will not be changed in Hsu's scheme might cause a false negative. As can be seen in Fig. 3.22, our lighting compensation scheme will still manage to adjust the color. To be more specific, the luminance of this image to a better level.




(a) (b)

*Fig. 3.22*(a) Original image. (b) Histogram of luma ($Y$) in $YC_bC_r$ color space

As to the average color criteria, for red tone images, Hsu's scheme might not work well either. For example, the image in Fig. 3.21 has a red tone background. The average color of that image is {85, 45, 32} (corresponding to the $R$, $G$, $B$ channels respectively) which falls inside the skin tone color boundary in our case. In contrast, Hsu's scheme will not change the image despite the fact that this image needs

lighting compensation to segment the human face from the skin-tone like background. With the aid of our adaptive lighting compensation, our scheme can segment the face and the background in the end. From the above, we can see that our scheme is robust enough to survive both the exceptions that Hsu's scheme does not handle well.

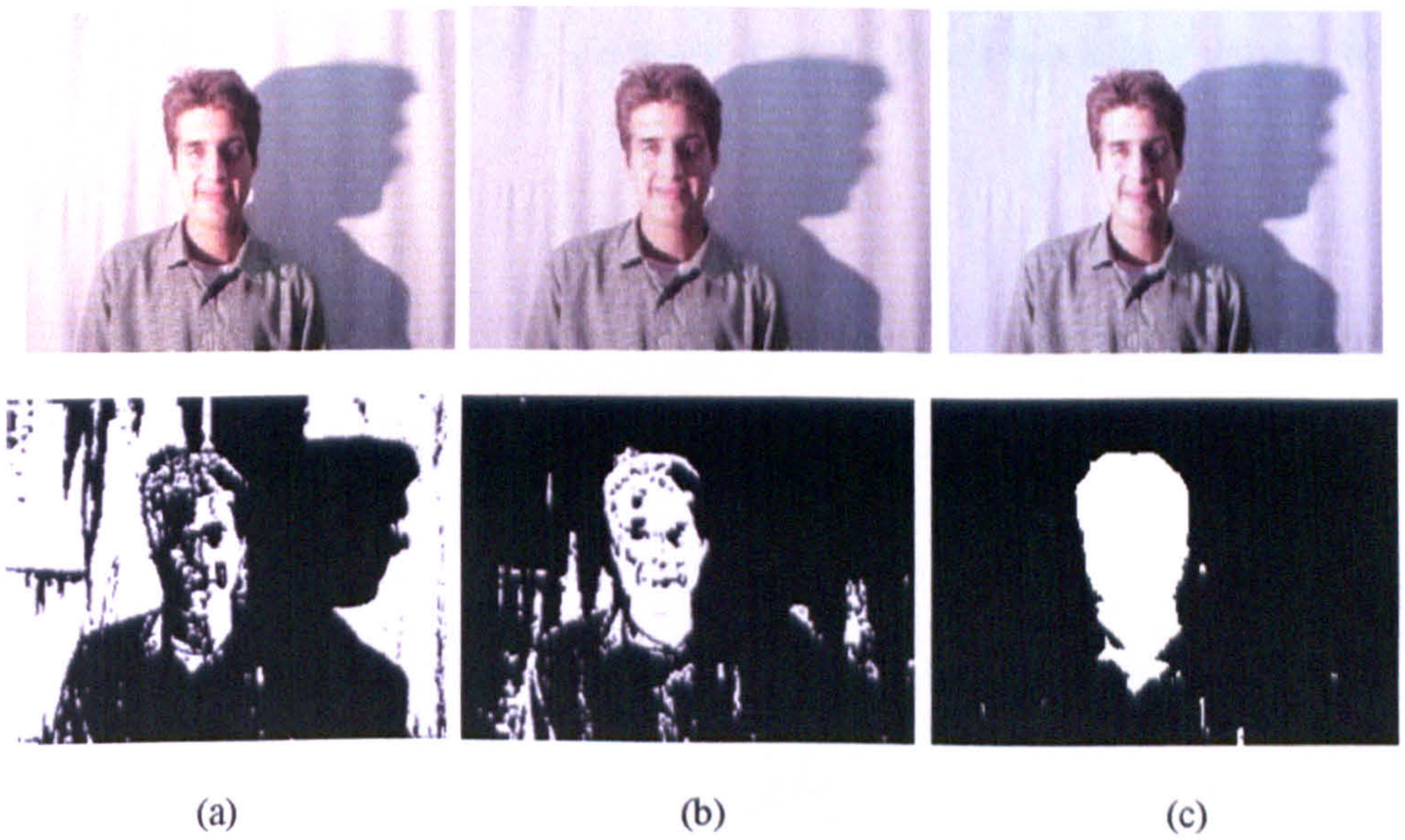



*Fig. 3.23* Image and mask gained after skin-tone color filter. (Image from HHI MPEG7 database [26]) (a) Original yellow-biased image. (b) Lighting compensation algorithm by Hsu et al. (c) Lighting compensation by the algorithm proposed.

As shown in Fig. 3.23, our adaptive lighting compensation method has a better correction result than Hsu's scheme. In Hsu's result, some parts of the background such as the curtain and the man's shirt are not corrected and remain yellow-toned. That causes some false positives after the skin color filter is applied. In our case, not only the number of false positives is much lower than Hsu's, but also the sizes of these patches are smaller, which can be judged from the visual result and the mask generated.

With respect to the skin color modeling, we know that there is a tradeoff between the False Positives (*FP*s), denoting non-face segments remaining in the image, and False Negatives (*FN*s), denoting real face segments missing from the image. The Detection Rate (*DR*) means the *FN* number ratio to all real faces. The stricter the

boundary for skin color, the less *FP*s and more *FN*s there would be. For automatic face detection security systems, having a few *FP*s is more tolerable than having an *FN*. But in reality, suppressing an *FN* usually means introducing hundreds or maybe thousands of *FP*s. With such a consideration in mind, we apply relatively loose restrictions on our skin color model to keep *FN*s as low as possible, and turn to the mask refinement for *FP* suppression. Results on the HHI MPEG-7 database are shown in Table 3.2 for a head on comparison with Hsu's work in a variety of head poses.

TABLE 3.2

DETECTION RESULTS ON THE HHI MPEG7 IMAGE DATABASE[76] (IMAGE SIZE 640X480)

| Head Pose | Frontal | | Near-Frontal | | Half-Profile | | Profile | | Total | |
|---|---|---|---|---|---|---|---|---|---|---|
| Schemes | Hsu | Ours | Hsu | Ours | Hsu | Ours | Hsu | Ours | Hsu | Ours |
| Image No. | 66 | 66 | 54 | 54 | 75 | 75 | 11 | 11 | 206 | 206 |
| Stage 1: Skin Color Filter | | | | | | | | | | |
| No. of FP | 3,145 | 18,253 | 2,203 | 14,992 | 3,781 | 222,481 | 277 | 3,059 | 9,406 | 58,786 |
| DR (%) | 95.45% | 93.94% | 98.15% | 98.15% | 96.00% | 97.33% | 100% | 100% | 96.60% | 96.60% |
| Stage 2: After Processing | | | | | | | | | | |
| No. of FP | 468 | 308 | 287 | 237 | 582 | 343 | 39 | 26 | 1376 | 914 |
| DR (%) | 95.45% | 93.94% | 98.15% | 98.15% | 96.00% | 97.33% | 100% | 100% | 96.60% | 96.60% |
| FP average | 7.09 | 4.67 | 5.31 | 4.39 | 7.76 | 4.57 | 3.55 | 2.36 | 6.68 | 4.44 |

FP: False Positive, DR: Detection Rate
Results measured on a PC with 1.7GHz processor

The experimental result is encouraging. The scheme we proposed achieves the same detection rate compared to Hsu's scheme. While at the same time, it manages to suppress more *FP*s, showing a 33.58% less *FP*s in average in the final result. Performance wise, as shown in Table 3.3, our mask refinement has a higher computation cost than Hsu's rectangle merging in stage 2. However, the advantage our scheme gained in stage 1 is significant. Thus it makes our schme 37.36% faster than Hsu's in overall computation time. Our second stage is slower simply because our morphological operators act on contours of patches for noise removal and smoothing and in this database this costs more operations than rectangle merging.

TABLE 3.3

PERFORMANCE ON THE HHI MPEG7 IMAGE DATABASE [76]

| Schemes | Hsu's | Ours | Diff. |
|---|---|---|---|
| No. of images | 206 | 206 | - |
| Time avg. in Stage 1 (sec) | 1.56 | 0.71 | 54.49% |
| Time avg. in Stage 2 (sec) | 0.18 | 0.39 | -116.67% |
| Total time average (sec) | 1.74 | 1.09 | 37.36% |

Results measured on a PC with 1.7GHz processor

Hsu also showed results on the champion database [77] from the Breakfast for Champions winners from 1999 to 2003. Since this database has been updated, we also used photos from the year 2004 in our test. In Hsu's case, 227 images were selected for testing but unfortunately it is not mentioned in the paper [18] how they did the selection. Thus, we choose to use all the images in the database to do a comparison, excluding only the non-photo and monochrome ones. Consequently, our image set contains 1267 images in total and it is guaranteed that the 227 images Hsu selected are included. The images in the Champion database are mostly portraits, ranging from frontal to half-frontal poses. Most of them are with simple background and simple variety of illuminations. The result of our scheme comparing to Hsu's is shown in Table 3.4.

TABLE 3.4

RESULTS ON THE CHAMPION IMAGE DATABASE [77] (IMAGE SIZE ~150X220)

| Scheme | Hsu | Ours |
|---|---|---|
| No. of Images | 227 | 1,267 |
| Stage 1. Skin Color Filter | | |
| No. of FP | 5,582 | 69,551 |
| DR (%) | 99.12% | 99.53% |
| Time avg. (sec) | 0.080 | 0.052 |
| Stage 2. After Processing | | |
| No. of FP | 382 | 1,801 |
| DR (%) | 99.12% | 99.53% |
| Time avg. (sec) | 0.012 | 0.047 |
| Final FP avg. | 1.683 | 1.421 |
| Total time avg. (sec) | 0.092 | 0.099 |

Results measured on a PC with 1.7GHz processor.
FP: False Positive, DR: Detection Rate

Performance wise, our scheme runs 7.6% slower on average than Hsu's. After analysis, we consider there are three causes for this result. First of all, the images in the Champion database have a smaller size which is beneficial for Hsu's scheme, while it is less advantageous for ours. This is due to the fact that in an image of $N$ pixels, Hsu's complexity is $O(N*logN)$ due to the sorting operation for "real white" pixel identification, while ours is $O(N)$. It can be seen that the distance $N*logN-N$ gets smaller as $N$ decreases, thus for small images, the speed-ups we gain are smaller. In fact, the performance gap between the two schemes in stage 1, which includes lighting compensation, color space transformation, and skin color filtering, is only 35% to our favor, as compared 54.49% in the HHI database.

Secondly, due to the simple background, the number of patches retained after stage 1 is much smaller than in the HHI database. In Hsu's scheme, distances of rectangle pairs are computed, thus for N rectangles the average time complexity is

$O(N^2)$. Our morphological operators act on each rectangle enclosed patch separately, so the time complexity is $O(N)$. It is obvious that Hsu's scheme would benefit more than ours from the simple background, since the distance between $O(N^2)$ and $O(N)$ in such cases is smaller due to the smaller number of rectangles to merge.

The last cause is that the sizes of the skin patches in the images are smaller. For the rectangle merging, the coordinates of the four vertices of a rectangle are dependent on the area of the patch. The coordinates are calculated by sorting the x and y coordinates of all pixels in the patch separately and then combining the minimum and maximum values from the two sorting operations. The sorting operation has an average time complexity of $O(N*logN)$, where $N$ is the number of pixels equal to the patch area in the discrete space. In our scheme, morphological operators work on the contour of the patch. Let M represents the number of pixels on the contour of a patch, it can be easily seen that M<=N, giving our scheme time complexity $O(M)$. Again we see that the reduction of the patch size benefits more Hsu's scheme than ours, since the distance $N*logN-M$ increases faster with $N$ than with $M$.

For the three reasons above, we can observe from the tables, that the advantage of the stage 2 in Hsu's scheme to ours increases from 116.67% to 291.67%.

Although lost ground in the performance comparison, our scheme still keeps the accuracy advantage. The average FP number of our scheme is 1.421, while Hsu's is 1.683. In this aspect, our scheme is 18.39% better. Furthermore, even with a much larger image set, our scheme can still achieve higher detection rate. As mentioned before, it is a tradeoff between accuracy and performance. We consider for an 18% accuracy increase, a 7.6% performance loss is worthy.

To further demonstrate the performance and accuracy of our scheme, experiments listed in Table 3.4 are run on 9 series (13 images each, 117 images in total, uniformed size 768x576) from the AR database [4] and dbf1 and Yahoo news photos databases (various sizes, 327 images in total) [24]. The AR database is designed for face recognition. All images in it have nearly plain white background and exactly one face in each image. Also, the images consist of frontal view faces

only. However, the difficulties of the AR database lie in the various facial expressions, different illumination, and occlusions (including sun glasses and scarves). The Yahoo News image database contains 327 real-life images. These images not only have complex backgrounds, but also contain more than one face per image. Furthermore, there are body parts other than faces, for example palms and arms in some of them. All these factors make it a very difficult database for face localization. The results are shown in Table 3.5.

TABLE 3.5

RESULTS ON AR DATABASE [79] AND YAHOO NEWS PHOTOS [78]

| Image Set | AR Database | Yahoo news |
|---|---|---|
| Image No. | 117 | 327 |
| Step 1. Adaptive Lighting Compensation | | |
| Time (sec) Average | 0.3685 | 0.1408 |
| Step 2. Skin Color Filter | | |
| No. of FP | 44,476 | 39,928 |
| DR (%) | 100% | 100% |
| Time (sec) Average | 0.8305 | 0.2925 |
| Step 3. Mask Refinement | | |
| No. of FP | 540 | 2,408 |
| DR (%) | 100% | 99.62% |
| Time (sec) Average | 0.5847 | 0.2505 |
| FP avg. | 4.6154 | 7.3639 |
| Runtime avg. (sec) | 1.7837 | 0.6838 |

Results measured on a PC with 1.7GHz processor
FP: False Positive, DR: Detection Rate

For the AR database, our scheme achieves a perfect detection rate, while the average FP number is well controlled to 4.6154. Visual result of a series of images is shown in Fig. 3.24. And on the Yahoo news database, our scheme can still achieve 99.62% detection rate, with the average FP number suppressed to 7.3639.

*Fig. 3.24* A set of images and results from AR database [29]

## 3.5 Conclusion

In this chapter, we have presented a fast skin region detector for color images. Our scheme first uses an adaptive lighting compensation method to correct the affection of illumination on color. Then a skin color filter within the normalized RGB color space is used to determine skin color pixels. Finally, morphological operators are applied in our mask refinement stage, in order to eliminate background noises, regain facial features, and smooth patch contours. The novelty in our scheme lies in the adaptive lighting compensation and the mask refinement with morphological

operators.

Experiments on several face image databases/photo galleries and comparisons to other works in the literature demonstrate the robustness and computational efficiency of our scheme. In next chapter, we will build up our face detector based on the work of our skin region detector.

CHAPTER 4

# Face Detector

## 4.1 Introduction

In this chapter, we will propose and analyze an unsupervised and robust face detector based on our previous skin region detector [92]. The skin region detector provided a solid foundation for the face detector that will be constructed in this chapter. But there are still quite a few false positives in the skin regions detected, including human skin regions other than faces and background objects which have colors close to skin. These false positives need further processing to be eliminated. Also, in order to localize the face and determine the orientation, the configuration of facial features has to be verified.

As mentioned in the literature review section, eyes are essential facial features needed to be detected for both face detection and recognition applications. Also as explained in the same section, eye corners are considered to be robust features to start the detection process with. Therefore, our face detector will start with detecting corners as interest points. But only eye corners are not yet enough to detect eyes, other cues like edges and local gray pixels are to be used later to verify the eye candidates. After the eye detection step, the mouth area is to be detected. Finally, configurational knowledge of human faces will be used to verify the face. A novel and efficient verification model is presented to verify the face candidate and determine the orientation of the face.

The chapter is organized as following: In Section 2, the construction of our eye detector will be shown. Follows up in Section 3, our mouth area detection and face verification process is presented. Experimental results and analysis would be given in Section 4. The chapter will conclude in Section 5.

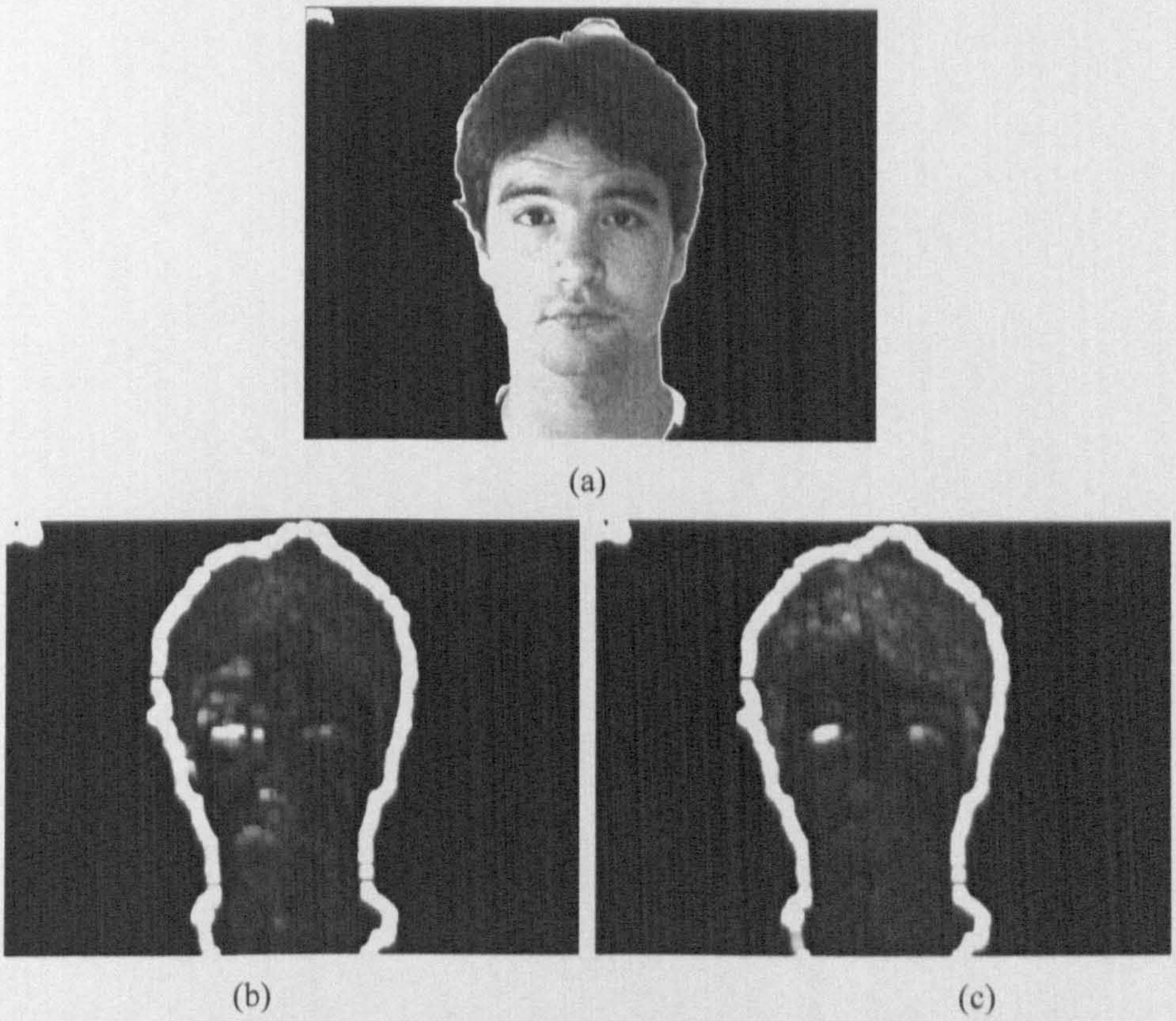
## 4.2 Eye Detection

The proposed scheme consists of four major steps:

- Adaptive corner detection
- Edge detection
- Local gray pixels detection
- Merging and eye localization

### 4.2.1 Adaptive Corner Detection

In the first step, a Harris corner detector [93] is used to detect corners in the skin patch. The assumption here is that human eyes form a special texture on the face image. In the normalized *RGB* color space, the red channel is mostly associated with skin color [92]. So that from the remaining two channels we chose the normalized blue ($N_B$ ) for eye detection because from experimental results, we find out that it is less prone to noise and illumination and gives better eye localization results than the normalized green channel ($N_G$ ). In Fig. 4.1, a face with uneven illumination is shown. When comparing 4.1(b) and 4.1(c), we find that for the right eye in shadow, the $N_B$ channel has a more distinguishable representation over the $N_G$ channel, as well as the noise level is lower.



(a) (b) (c)

*Fig. 4.1.* (a) Original image (b) Dilated image of $N_G$ channel (c) Dilated image of $N_B$ channel

In the traditional Harris corner detector, the image derivatives $d_x$ and $d_y$ are first calculated. Then the squared derivative $d_x^2$ and $d_y^2$, together with $d_x d_y$ are smoothed by a Gaussian filter, which in turn result in the Gaussian blurred derivative image $I_{xx}$, $I_{yy}$ and $I_{xy}$. After this, the Harris measurement is computed as (4.1).

$$(I_{xx} \bullet I_{yy} - I_{xy}^2) - 0.04 \times (I_{xx} + I_{yy})^2 \tag{4.1}$$

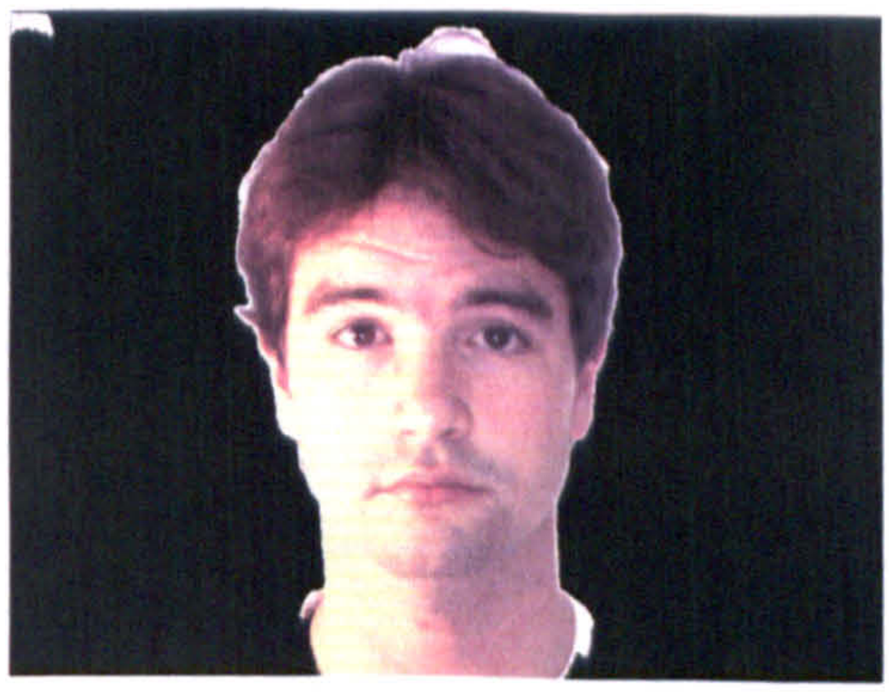
Finally, local maxima are extracted by performing a grey scale morphological dilation and then finding points in the corner strength image that match the dilated image and are also greater than a threshold.

The traditional Harris corner detector is rotation-invariant, but is not scale invariant. Multi-scale version Harris corner detector is suggested in [98][99]. According to K. Mikolajczyk et al's comparison result in [100], the Harris corner detector has the second matching score, which the Hessian-affine descriptor has the best score. But the computation cost of the Hessian matrix is much higher than the

Harris corner detector. As Raghavendra et al pointed out in [101], the calculation of the Hessian matrix requires 36 addition and 36 multiplication operations for each pixel. This is too high a cost for us to afford in our system.

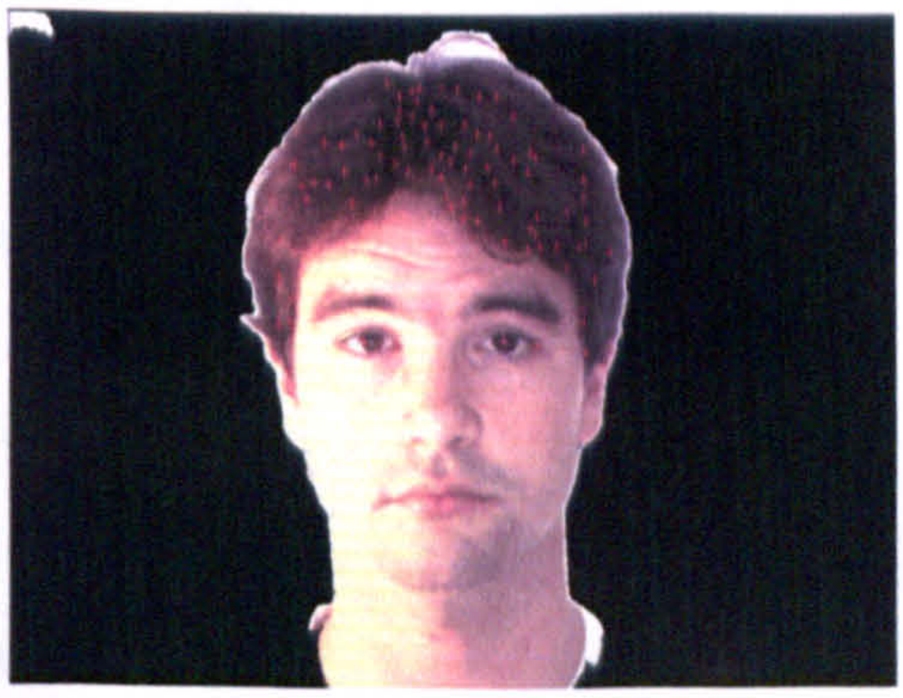
Other point of interest detectors, such as SIFT [102] and SIFT-PCA [103], are introduced and reviewed in [104], but their accuracy and robustness improvements are based on dramatically increased computation cost. At the same time, it is not guaranteed to give us the local interest point needed for the eye corner detection.

For example, we found out that the SIFT algorithm performed even worse than the Harris corner detector after we tried it on some of images from the Purdue AR database [79] and HHI MPEG7 database [76]. Given the efficiency and robustness of the Harris corner detector shown in the experiment, we consider it good enough for our eye corner detection scheme. Two examples of the failure of SIFT detector are shown in Fig. 4.2. From the examples, it is easy to tell the SIFT detector is not very suitable for detecting eye corners locally. In Fig 4.2(a), missing an eye corner in a skin patch with high resolution is not acceptable. What is more, if we compare Fig 4.2(b) and Fig 4.3(a), we can see the SIFT detector is more sensitive to the irregularity at the contour of the skin patch. The reason here is that the SIFT detector is based on finding the local maxima and minima of the Differentiation of Gaussian (*DoG*) considering the 26-neighbors in three layers of *DoG*. But the Gaussian convolution is operated on the global image, and the $\sigma$ used is fixed. Without considering the contour of the skin patch, the irregular points at the contour with high contrast to the background are easier to become the local maxima.



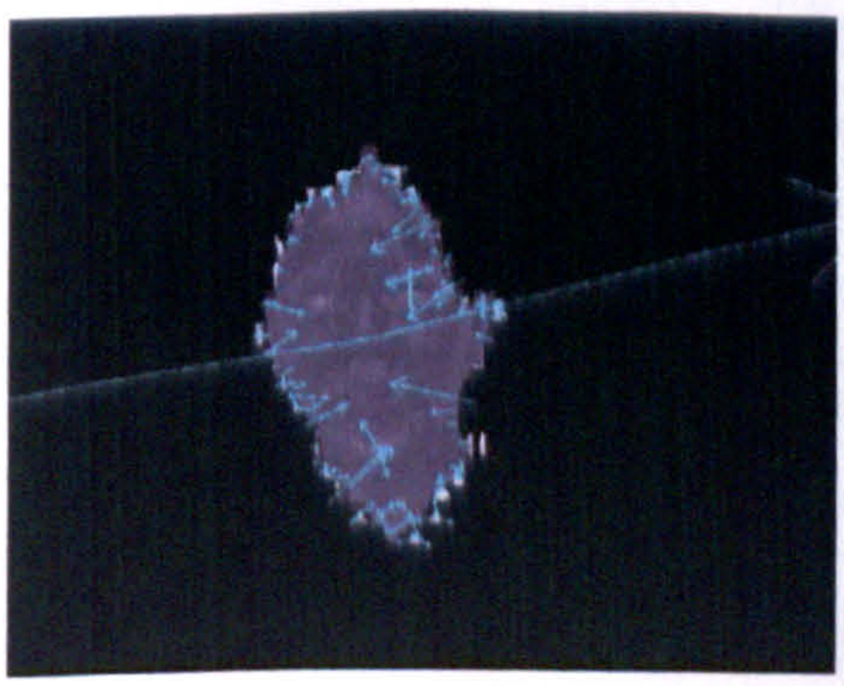
(a)

(b)

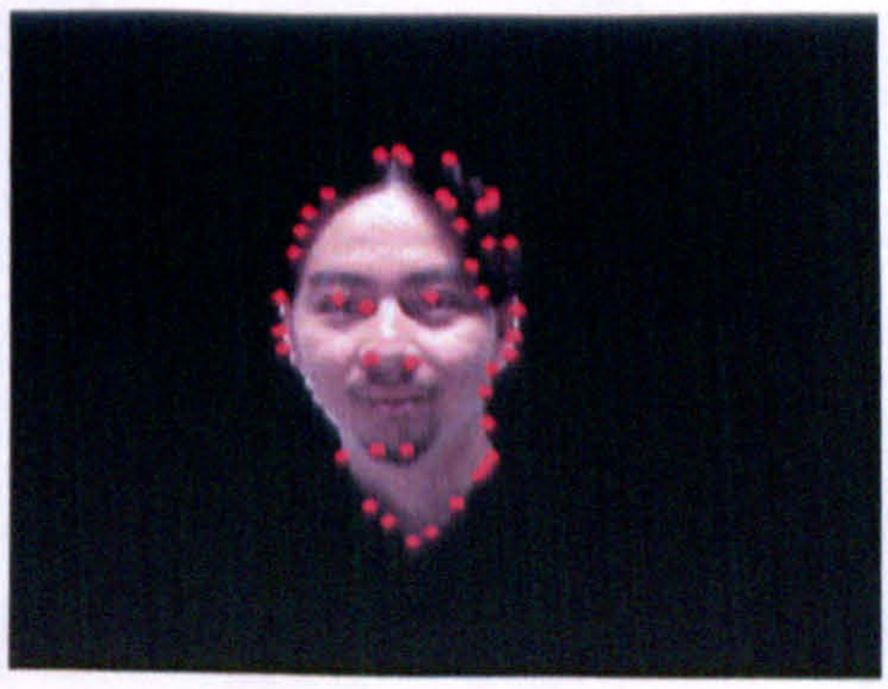


(c)

(a)



(b)



(c)

*Fig. 4.2.* Detector comparison. (a) original image (b) result of SIFT detector (c) Harris corner detector

Considering the eye in the shadow problem, a low threshold value is applied to the Harris corner detector to guarantee that local maxima will not be missed, at the potential expense of many false positives. Also, to handle the scale problem, we make the standard deviation of the smoothing Gaussian adaptive to the size of the skin patch with the condition shown in (4.2).

$$\begin{cases} \sigma = 1, \max(x, y) < 200 \\ \sigma = 2, 200 \le \max(x, y) \le 400 \\ \sigma = 3, \max(x, y) > 400 \end{cases} \tag{4.2}$$

Given that σ is the standard deviation of the smoothing Gaussian, and $x$, $y$ present the width and height of the skin patch respectively. The following assumption is made: The higher the resolution of the possible face, the more details would be contained within the patch, thus the larger size of the kernel window should be used for smoothing. This means that more trivial details are to be smoothed out.

(a)

(b)

(c)

*Fig. 4.3* Edge points removal (a) Harris corner detected (b) Edge map of the mask (c) After removal

From the experiments, we find out that most of the false corners appear on the boundary of the skin patch, as shown in Fig. 4.3 (a). This is due to the high contrast and at between skin colors and the black background and the non-convex edge of the skin patch. Then we start eliminating these false positives from the boundaries of our skin patches by initially detecting edges on the borders of the patch using the Sobel edge detector. Since the computation of the edge is performed on a binary image, it is very fast (less than 0.3s for a 768x576 image from the AR database [79]).

After that, we thicken the edge by morphological dilation with a diamond-shaped structure element. Then we gain an edge mask of the skin patch (Fig. 4.3 (b)), and then it is used to remove the corners which fall inside that mask (Fig. 4.3 (c)).

### 4.2.2 Local Gray Pixels Detection

As we can see from Fig 4.2, other facial features such as facial hair, nostrils or mouth can also have corners considered as interest points by the Harris corner detector. The next step is to detect local gray pixels. The assumption here is that eyes contain local gray pixels.

This is based on three arguments. One is that, we all know that there are white parts on eyeballs, which would appear as 'gray' in the normalized *RGB* color space we are using. Another one is that eyelashes and eyebrows are usually of dark color. Last but not least, even when the eye is under the shadow caused by the nose bridge or other objects, the iris without reflection will appear as dark color. So no matter if the eyes are highlighted or belong to a shadow area, the darker eyeball and the white of the eye will remain relatively gray with respect to adjacent skin pixels. These pixels would appear locally 'gray' in the image. In the normalized *RGB* color space, we consider pixels with value:

$$|N_G - 0.333| + |N_B - 0.333| \leq 0.09 \tag{4.3}$$

as local gray values.

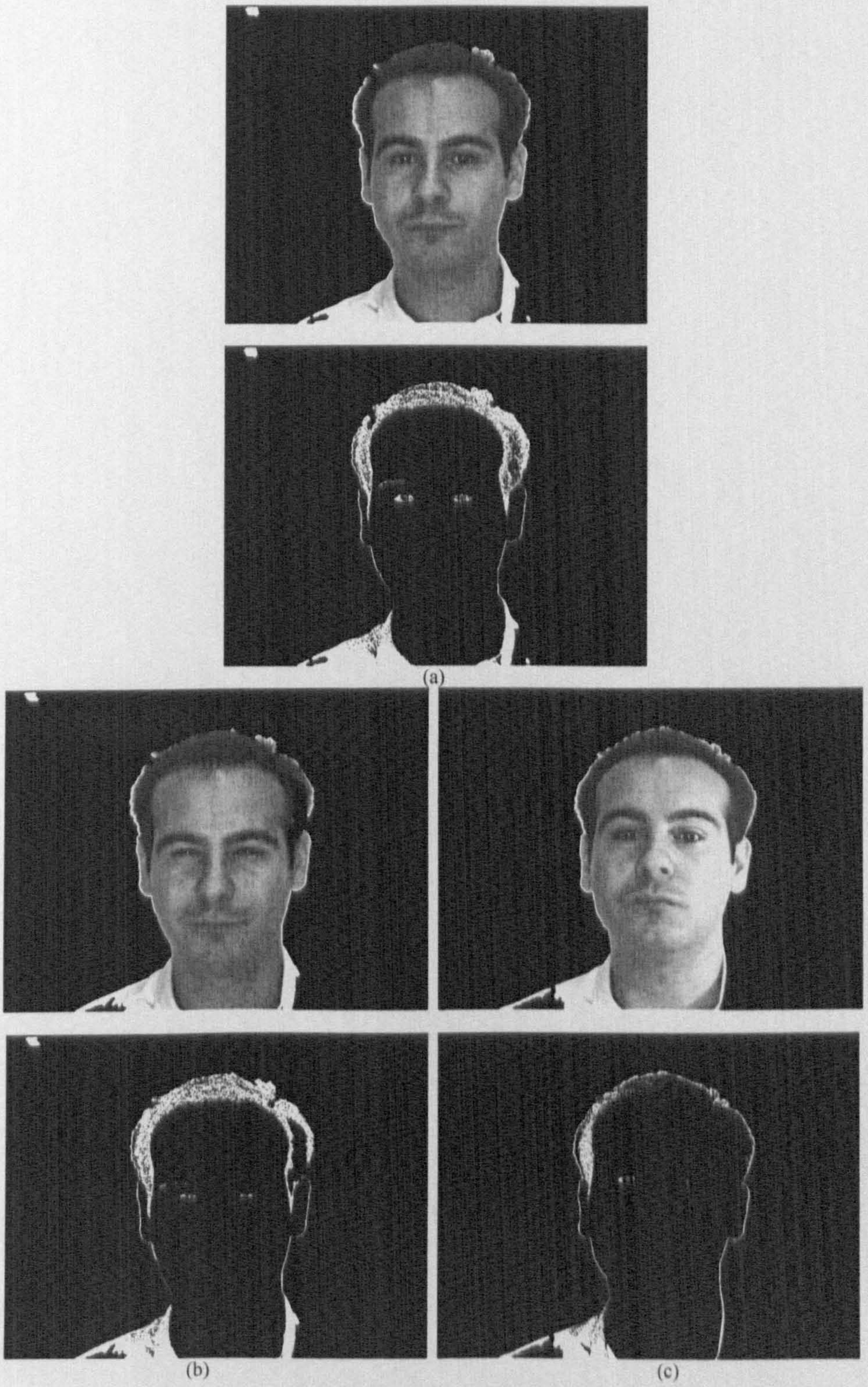



*Fig. 4.4* Local gray pixels change when expression and illumination changes (a) Ordinary image (b) Expression change (c) Illumination change

A gray map is then built and smoothed by a gray-scale close filter to merge close regions, remove outliners, and smooth edges. The local gray pixel detection applied

to a series of the AR Database [79] is shown in Fig. 4.4. From Fig 4.4, we can see that the local gray pixel of the eye area is robust to both expression and illumination. In Fig. 4.4(b), the eyes are narrowed due to expressions, and in Fig. 4.4(c), the eyes are under uneven illumination, one is under bright lighting while the other is in shadow. Yet in both cases, there are still local gray pixels detected.

After that, the gray map is combined with the surviving Harris corners from the first step that are located inside the patches. Only corners matches the local gray pixels around will be kept and passed to the next step. As shown in Fig. 4.5, the two corners caused by nostrils can now be removed because there is no local gray pixel adjacent to them. There are still two remaining FPs because the black beard also shows as relative gray. In our practical implementation, in order to save computation, we combine the gray map with the edge map gained in next step, before start eliminating false positives.

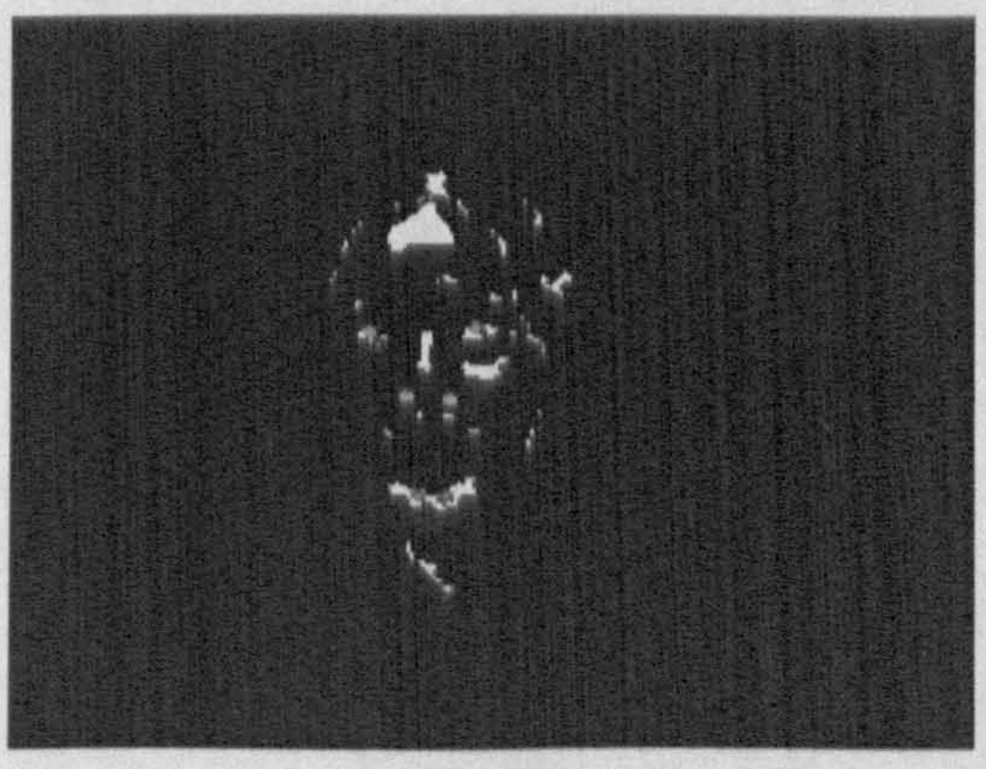


*Fig. 4.5.* Gray map

### 4.2.3 Edge Detection

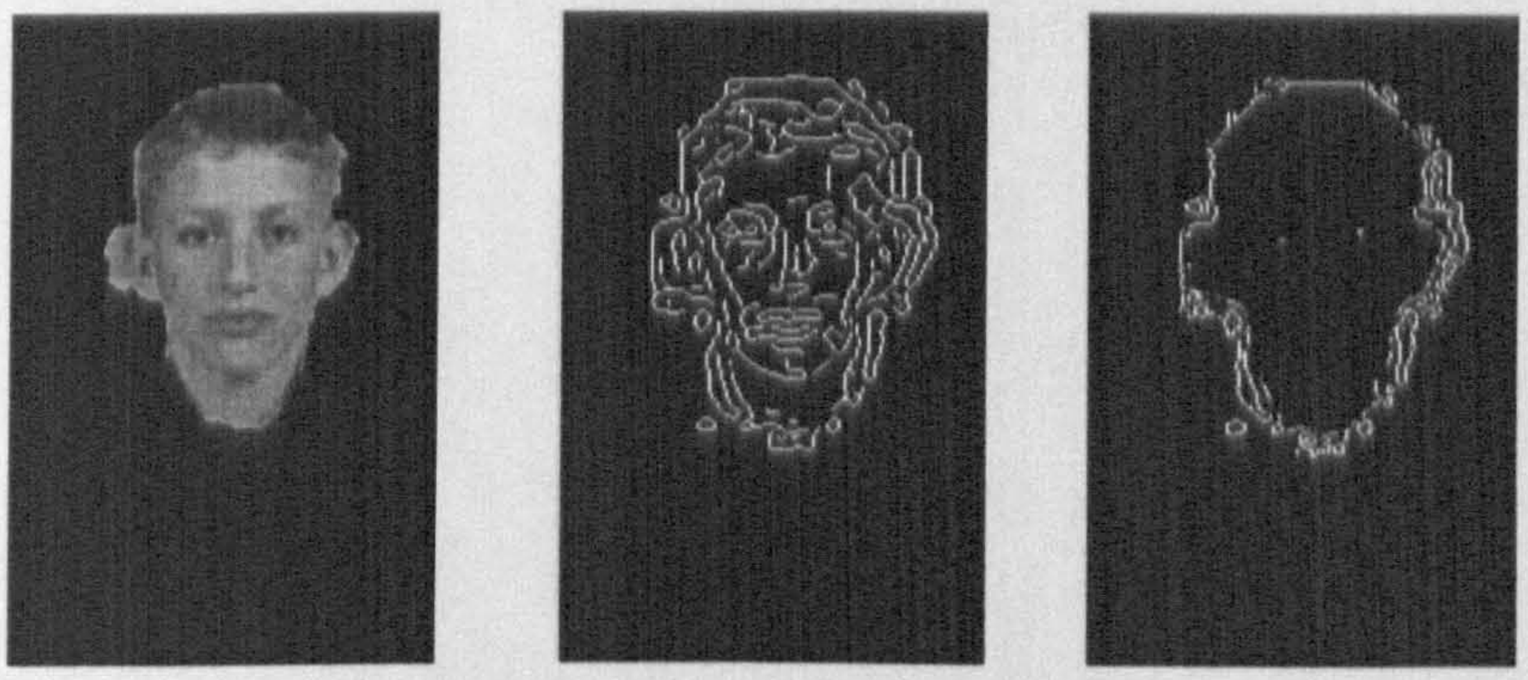
The third step of our scheme is edge detection. Different edge detectors are used to generate the edge map of the skin patch.

The argument here is that the special structure of the eye would show many edges on the skin patch. The eye socket, the eyeball, the eyelid, the eyelash, the eyebrow, and even wrinkles at the eye corner would generate identifiable edges

around the eye corner detected. Some edge detection algorithms are introduced and evaluated in [105]. As shown in the experimental results of the [105], The Canny edge detector [106] achieved the best detection rate when applied for detection applications, and has the second lowest computation cost in the five representative edge detection approaches.

From empirical experience, we addressed a major disadvantage of the Canny edge detector for our application. The problem is that it would generate too much edges for a high resolution patch because the richness of details. At the same time, the performance suffers. The problem is illustrated in Fig 4.6. The reason of the overkill of Canny detector is the edge tracing step. For images with low resolution, Sobel detector is good enough. At the same time, Sobel detector is essentially one single step of the Canny detector, so that the performance will be improved significantly.

For skin patches with dimension $\max(x, y) \leq 400$, a Canny edge detector is used. Otherwise, a Sobel detector is used. The reason here is that, for small size skin patches, the Canny edge detector can retain more details inside the patch, which is more important to detect a low resolution eye. On the contrary, for a large size skin patch, a Sobel detector is insensitive to noises.



(a) (b) (c)

(d)

(e) (f)

*Fig. 4.6.* Edge detector comparison for images: (a)(d) Original images, (b)(e) Results of Canny detector (c)(f) Results of Sobel detector

From Fig. 4.7, we can see that with the help of the edge map, the two false positives caused by the beard can now be eliminated in the upcoming merging and eye detection step.

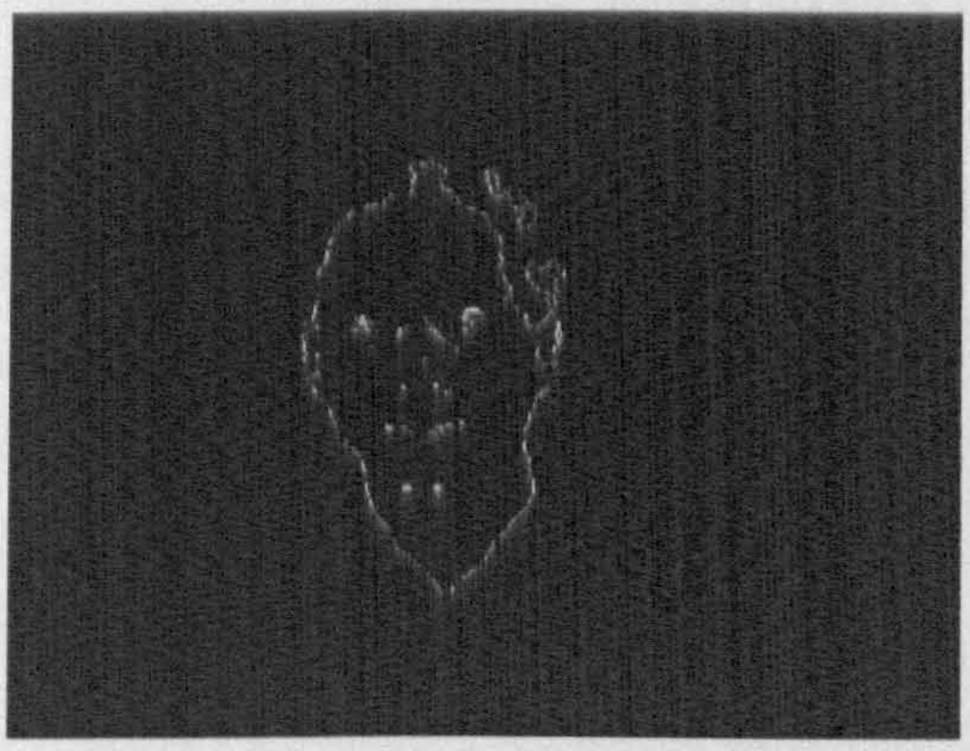


*Fig. 4.7*. Edge map

### 4.2.4 Merging and Eye Detection

The final step of our eye detection scheme is to group the internal Harris corners to blocks, whose size is also adaptive to the size of the skin patch. Since the contents of these blocks are indeed binary values, their sum is calculated and the block can be accepted or rejected by a threshold. In our current experiments, all non-zero sum blocks would be retained. This computation is accelerated by a convolution operation with 3-by-3 matrix made of ones.

Then a Gaussian smoothing operation is used and local maximas are extracted. These also help to merge adjacent interest points. The final result of the process is shown in Fig. 4.8.

*Fig. 4.8* Final result of the proposed eye detector

Another example of the whole processing of our eye detector is given in Fig. 4.9.

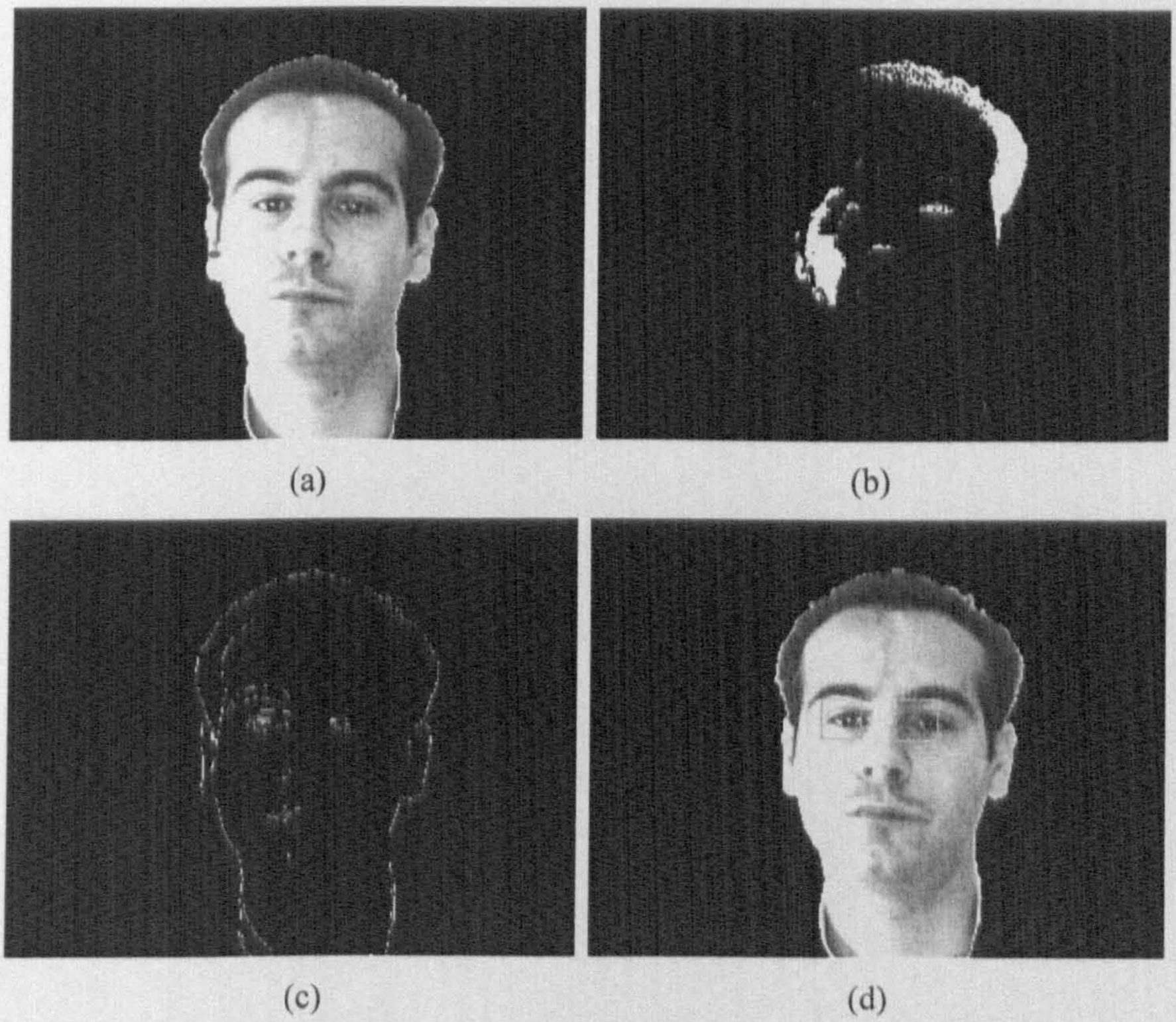
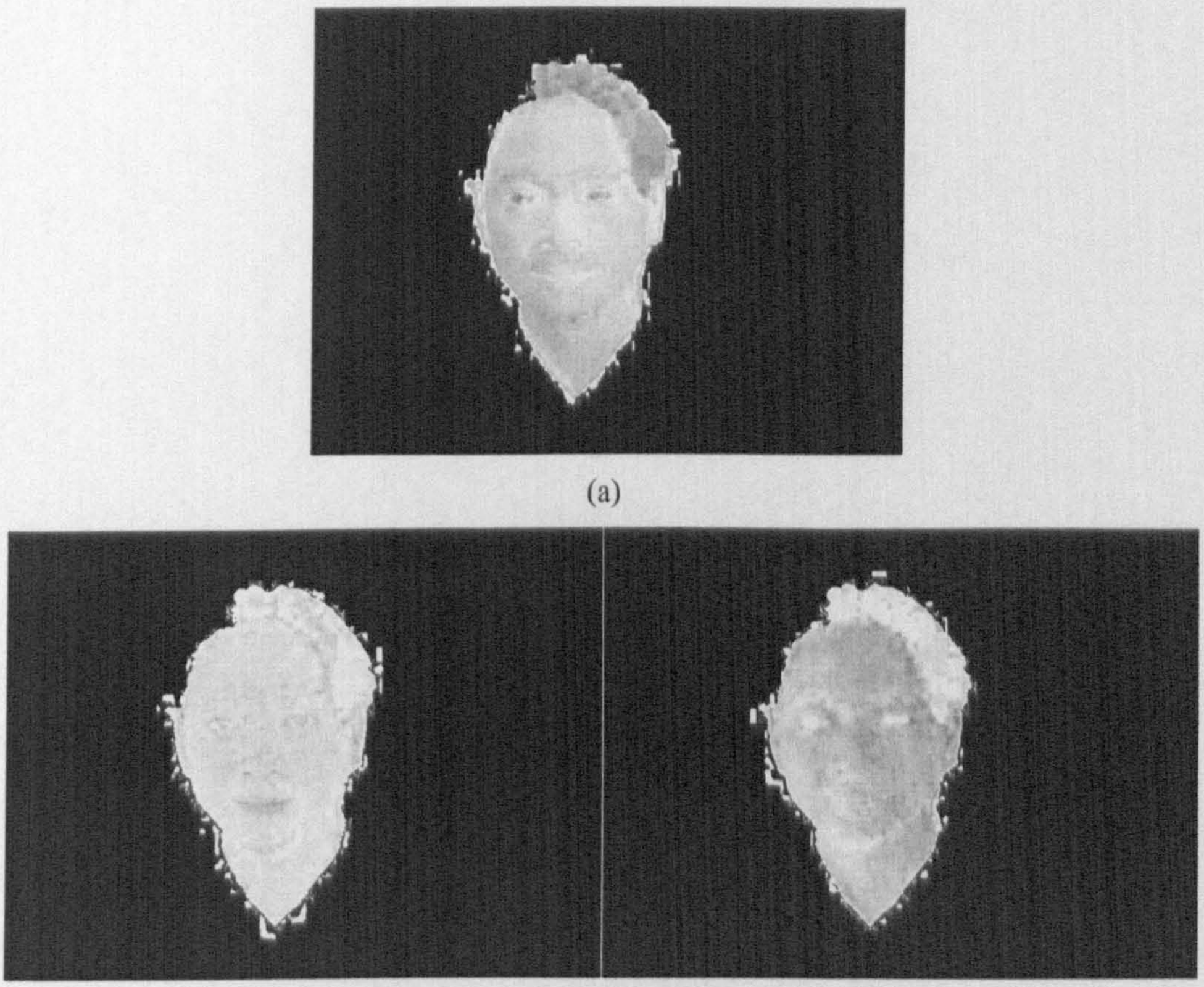


*Fig. 4.9* (a) original image (b) gray map (c) edge map (d) final result

## 4.3 Face Verification

### 4.3.1 Mouth Area Detection

Similar to the eye detector, we first observe the characteristics of the mouth in different channels. The empirical results show that in the $N_G$ channel, our mouth area detector works better than in the $N_B$ channel. As can be seen from Fig 4.10, in the $N_G$ channel, the mouth area is distinctive and other features like eyes and facial hair are obscured. Therefore, the $N_G$ channel would be used to detect the mouth.



(a)

(b) (c)

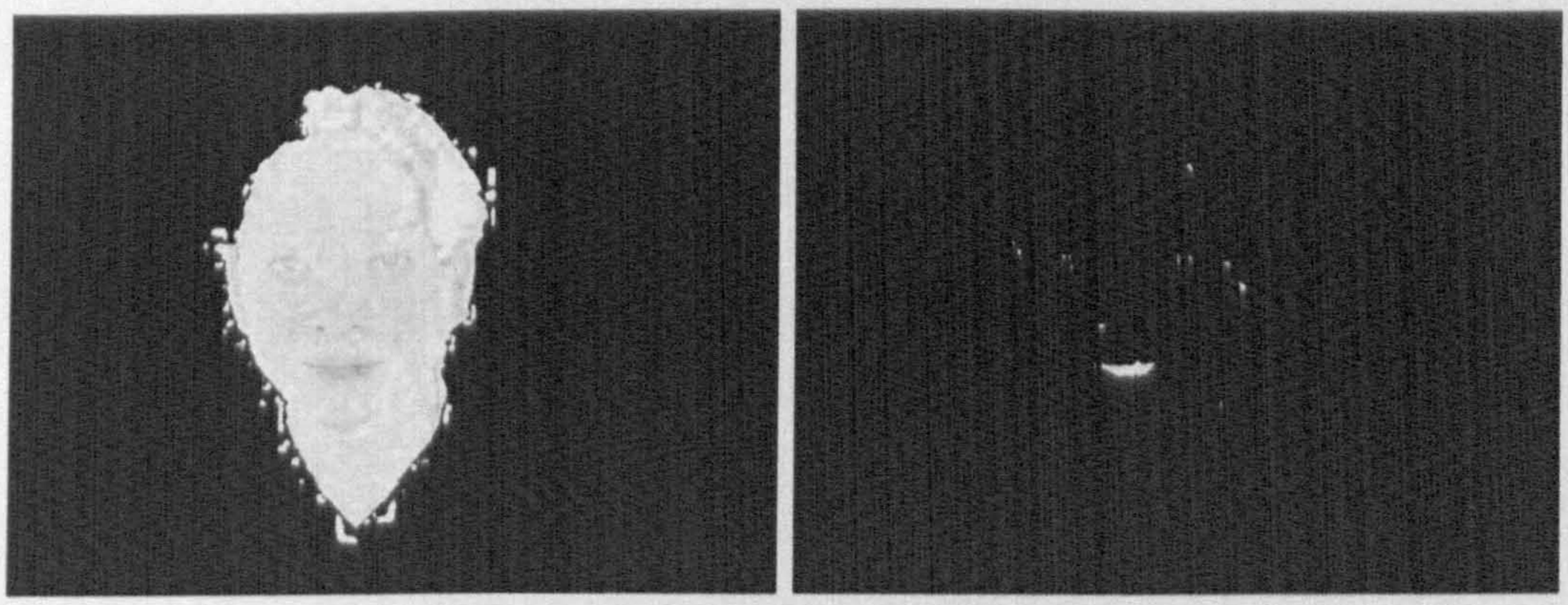
*Fig. 4.10* The mouth area in different channels (a) $N_R$ channel (b) $N_G$ channel (c) $N_B$ channel

First of all, the $N_G$ channel of the skin patch is stretched to full range. This helps increasing the contrast between the mouth area and the adjacent skin. The range to be stretched is defined by the bottom 1% and the top 1% of all pixel values. A histogram would be built with 100 bins. Then the average value ($d_{low}$) of the first bin and the last bin ($d_{high}$) would be used as the current range value. Pixels with value under $d_{low}$ or over $d_{high}$ would be clipped. Finally the range is expand to the full range [0, 1] by (4.4):

$$d = \frac{d - d_{low}}{d_{high} - d_{low}} \tag{4.4}$$

After the range stretch operation, a gray-scale morphological opening operator with a diamond-shaped structure element is used to smooth the skin patch, in order to remove minor noises. The result of this process is shown in Fig 4.11 (a).

Finally, the average of the non-black pixels is calculated. The pixels higher than the average would be filtered out. Also, considering the effect JPEG-artifacts, we need to eliminate the pixels on the edge of the patch, using the mask edge generated in the eye detection process. After filtering out the pixels, again we apply a morphological opening operator on the image to eliminate the outliners. The final result is shown in Fig 4.11(b).

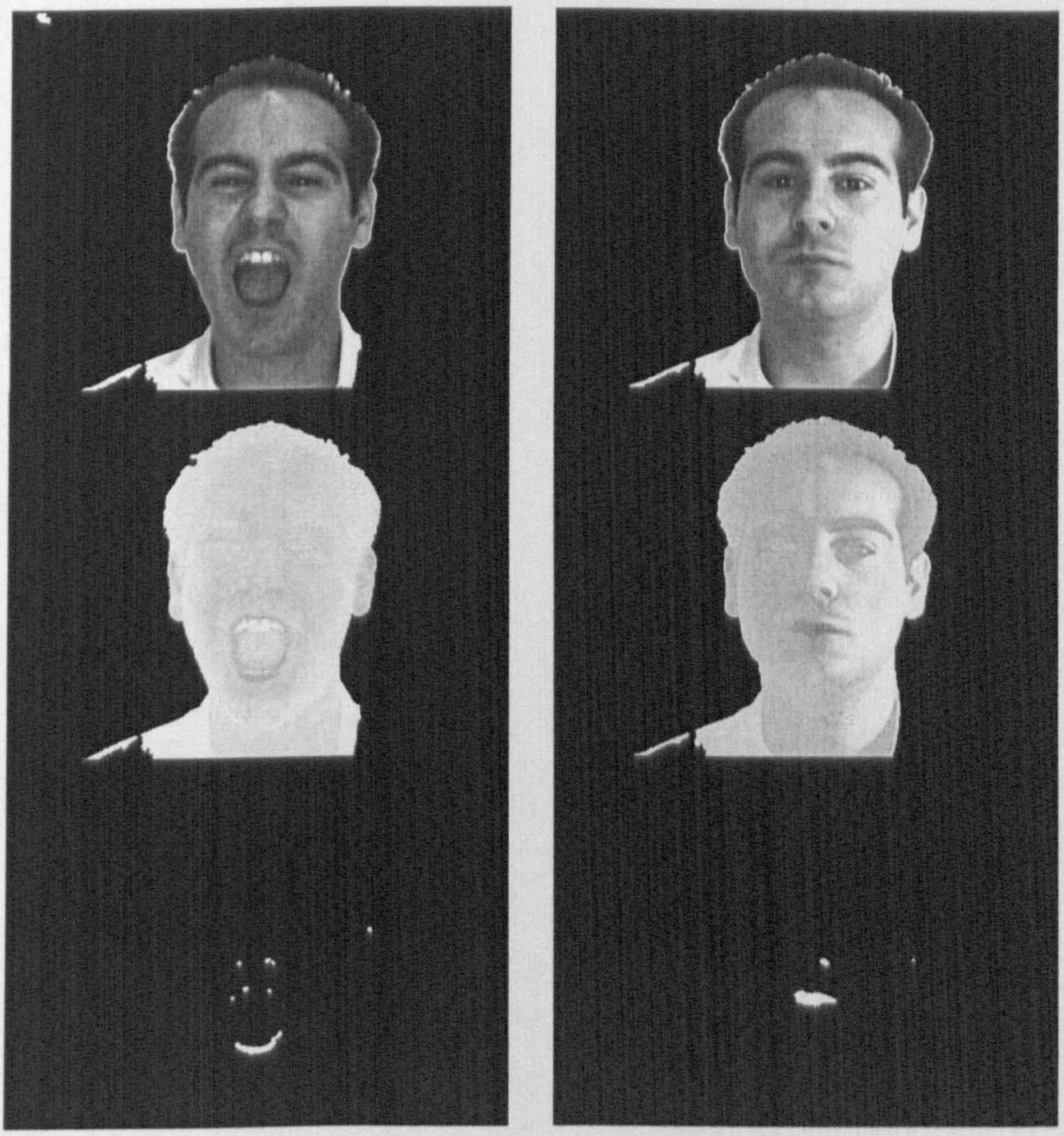


(a) (b)

*Fig. 4.11* Mouth area detection (a) smoothed image (b) mouth area detection

After some experiments, we found out that our mouth area detection approach is robust to both illumination and expression. This is illustrated in Fig. 4.12. It is noticeable from the Fig. 4.11(b) that given high enough resolution, the nostril would also be detected by the detector. These false positives of nostrils will not interfere the verification process in the upcoming section, on the contrary, they help making the verification process more robust.

We choose not to call our mouth area detection process a mouth detector. The reason here is that it is not designated to detect the mouth. Nostrils, eyeballs and shadows on the face might also be detected, although they might not at all be able to interfere in the upcoming configurational verification. Not to cause confusion about the definition of this detection process, we consider it an integrated step in our face verification scheme.



(a) (b)

*Fig. 4.12* Mouth area detector under expression and illumination changes (a) Ordinary image (b) Expression change

### 4.3.2 Configurational Face Verification

As mentioned in Chapter 1, human face perception skill is based on both feature and configurational information. So that based on the eye candidates and the mouth area detected in the last two sections, we start our face verification process based on the configurational knowledge of human faces.

For each pair of eye candidates detected in section 4.2, a verify area is generated within a rectangle, as shown in Fig. 4.13. The two circles represent the detected eye

candidates. The angle between the two diagonals is 30 degrees. The two diagonals define the verification area for the mouth area. Since we do not assume the orientation of the face, the verification area contains two possible sides. The orientation of the verification area would be decided by the location of the eye candidates and the size proportional to the distance between the two eye candidates. So that it is both scale and rotation invariant.

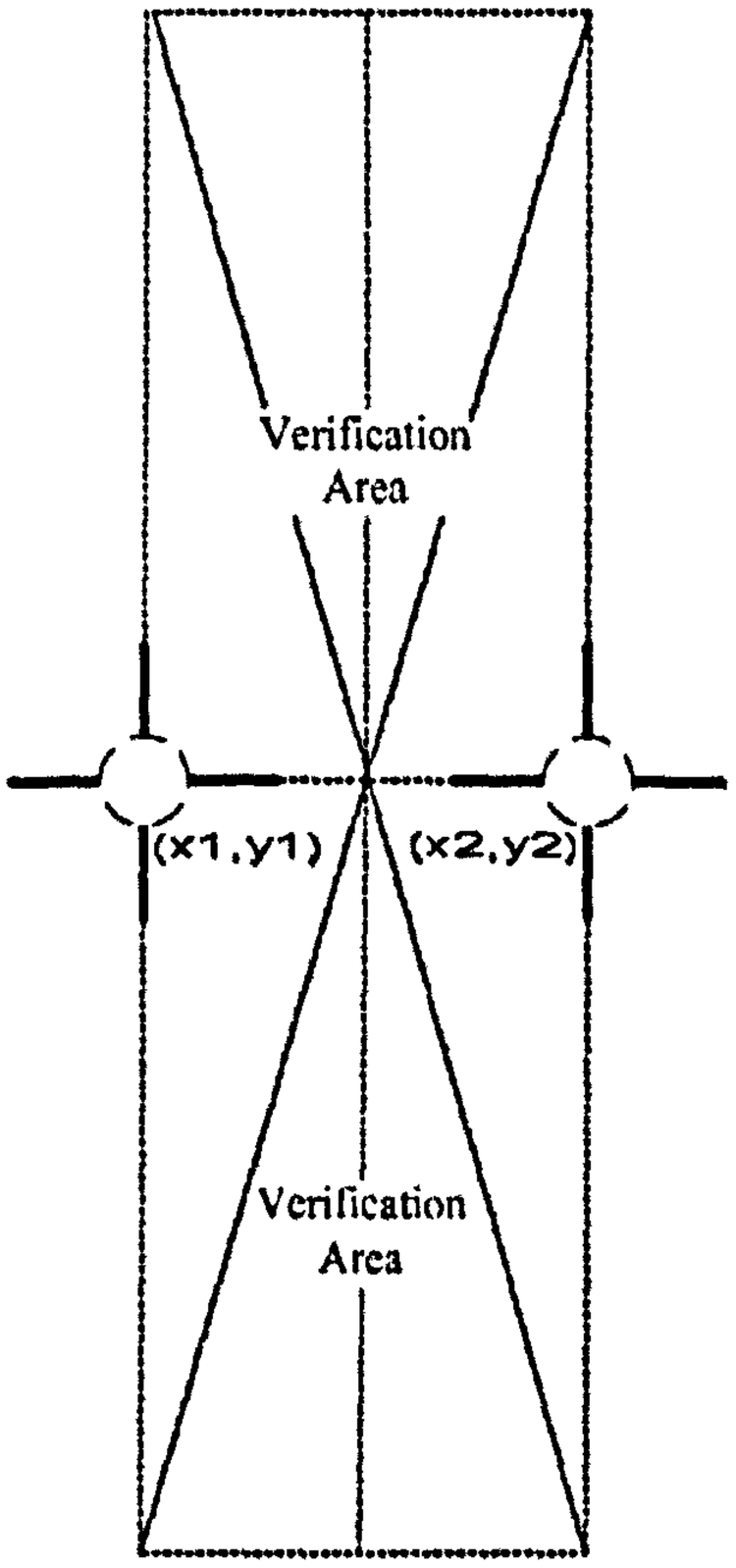


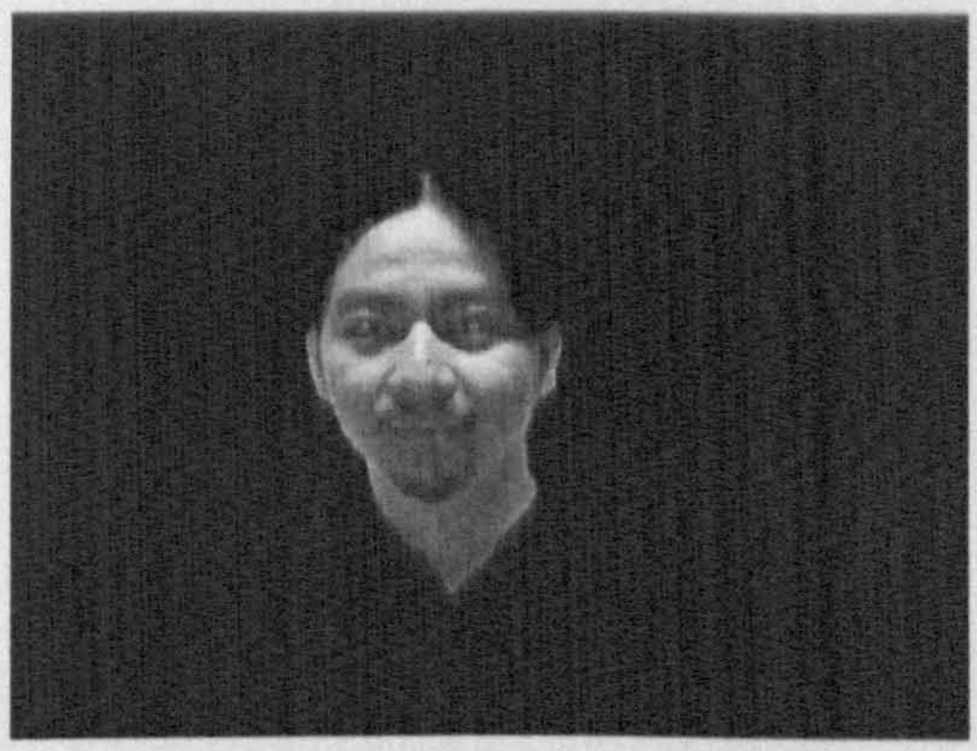

*Fig. 4.13* The verification area.

Given a pair of face candidates with location $(x_1,y_1)$ and $(x_2,y_2)$, the vertexes of the verification area can be decided as:

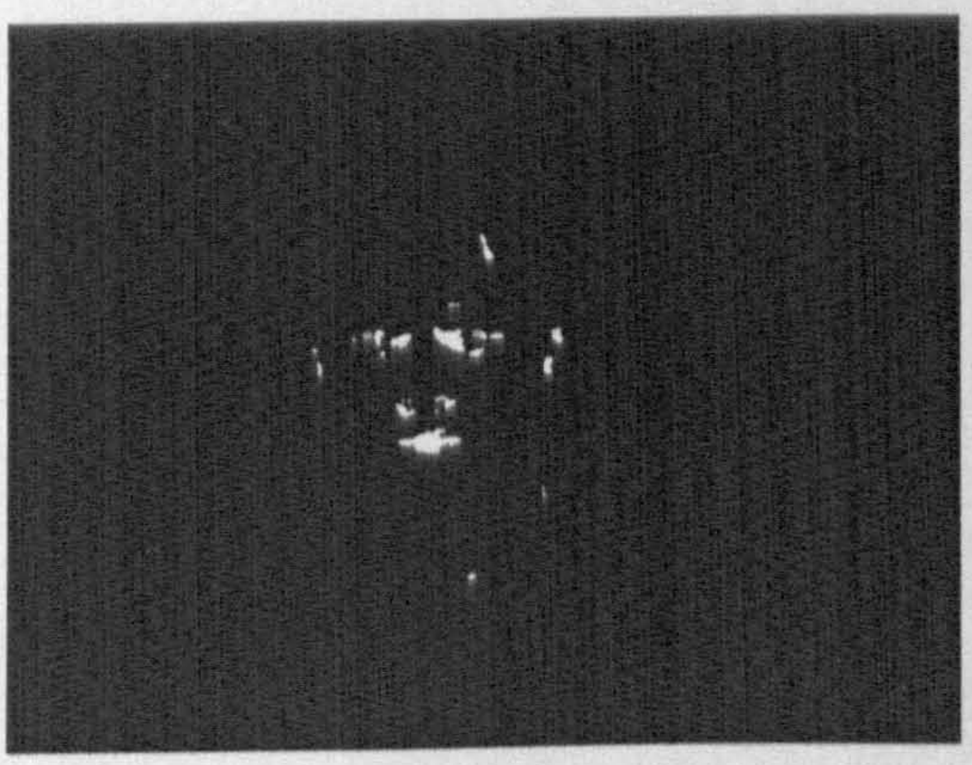
$$
\begin{aligned}
D &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\
D_L &= D \cdot \text{cotangent}(15^{\circ}) = D \cdot (2 + \sqrt{3}) \\
\Delta_x &= \frac{|y_1 - y_2| \cdot D_L}{D} \\
\Delta_y &= \frac{|x_1 - x_2| \cdot D_L}{D}
\end{aligned}
\tag{4.5}
$$

The five vertexes are then with coordination $(x_1-\Delta_x, y_1-\Delta_y)$, $(x_2-\Delta_x, y_2-\Delta_y)$, $(x_1+\Delta_x, y_1+\Delta_y)$, $(x_2+\Delta_x, y_2+\Delta_y)$ and $\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$. All the values would be rounded because the coordinates in digital images are discrete integer values. Also, values exceed the boundaries of the patch would be clipped using the mask.

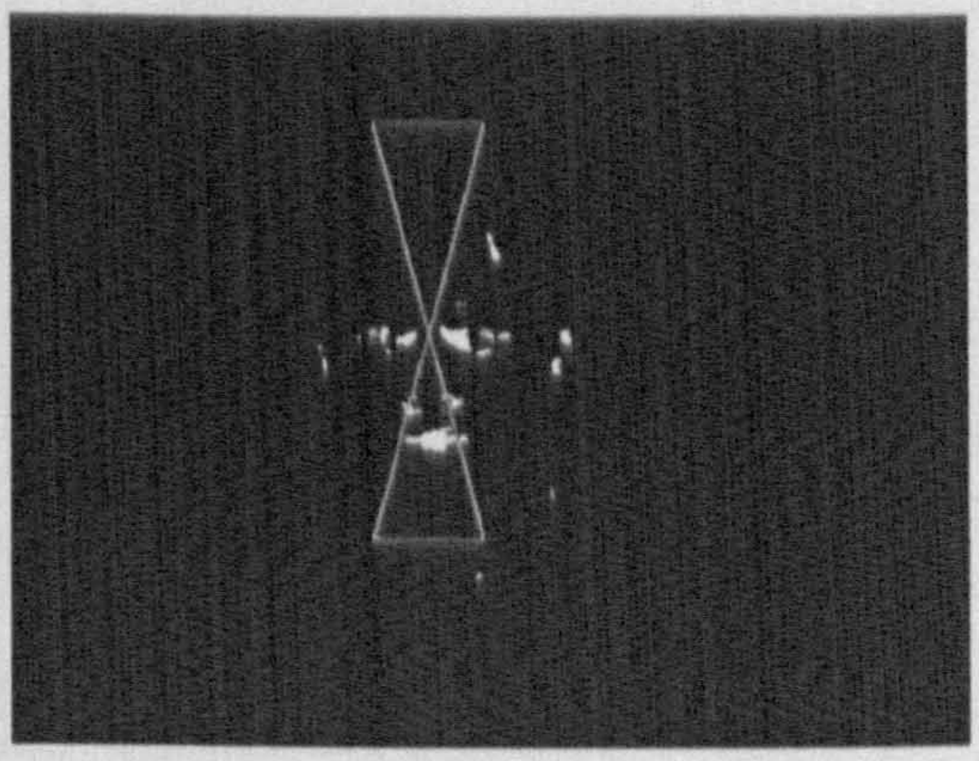
If there is mouth area detected within the verification area, the eyes and the mouth are considered as a possibly correct configuration of a face. If there is none, the pair of eyes candidates is not a correct pair of eye candidates for verifying the face. Illustrated in Fig. 4.14, with the eye candidates detected, the verification area is decided. The verification area successfully evades the interference of the false positives caused by ears, eyes and shadows on the face in the mouth area detection procedure. On the other hand, false positives caused by nostrils, would in reverse strengthen the accuracy of the verification.



(a)



(b)



(c)

*Fig. 4.14* The verification process. (a) Eye detection (b) Mouth area detection (c) Verification

False positives of eye candidates and mouth areas remain an issue for our face verification process. In some cases, more than one eye candidates for each eye are detected. Therefore, some rules are needed to handle the effect of false positives from both the previous eye detection and mouth area detection, and also duplicate eye candidates. The rules are listed as following:

(1) For two overlapping rectangles that both have mouth areas inside, the one that has more mouth area pixels wins.

(2) Since the verification area has two sides, in the case of mouth area appears in both sides, the side has the largest area is consider as the side for face configuration.

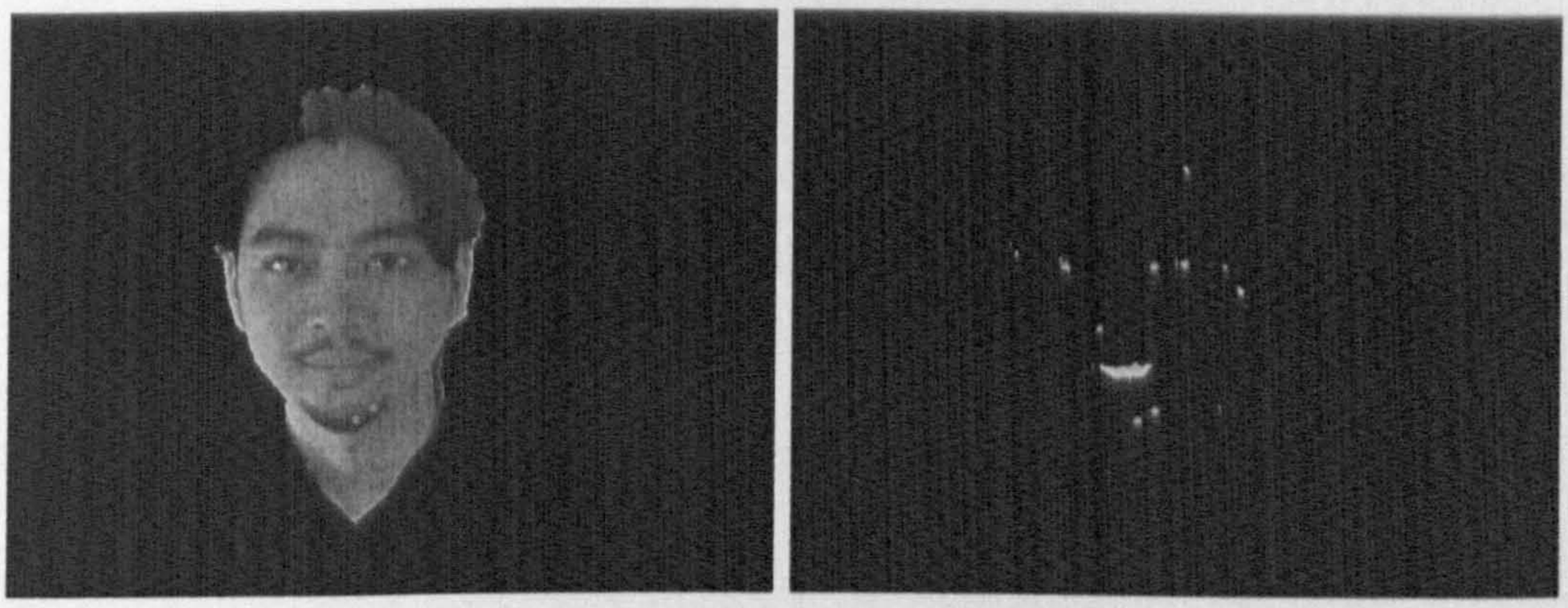



*Fig 4.15* Verification Rules (a) Eye candidates (b) Mouth area detected (c) Verification areas (d) Rules are needed for verification

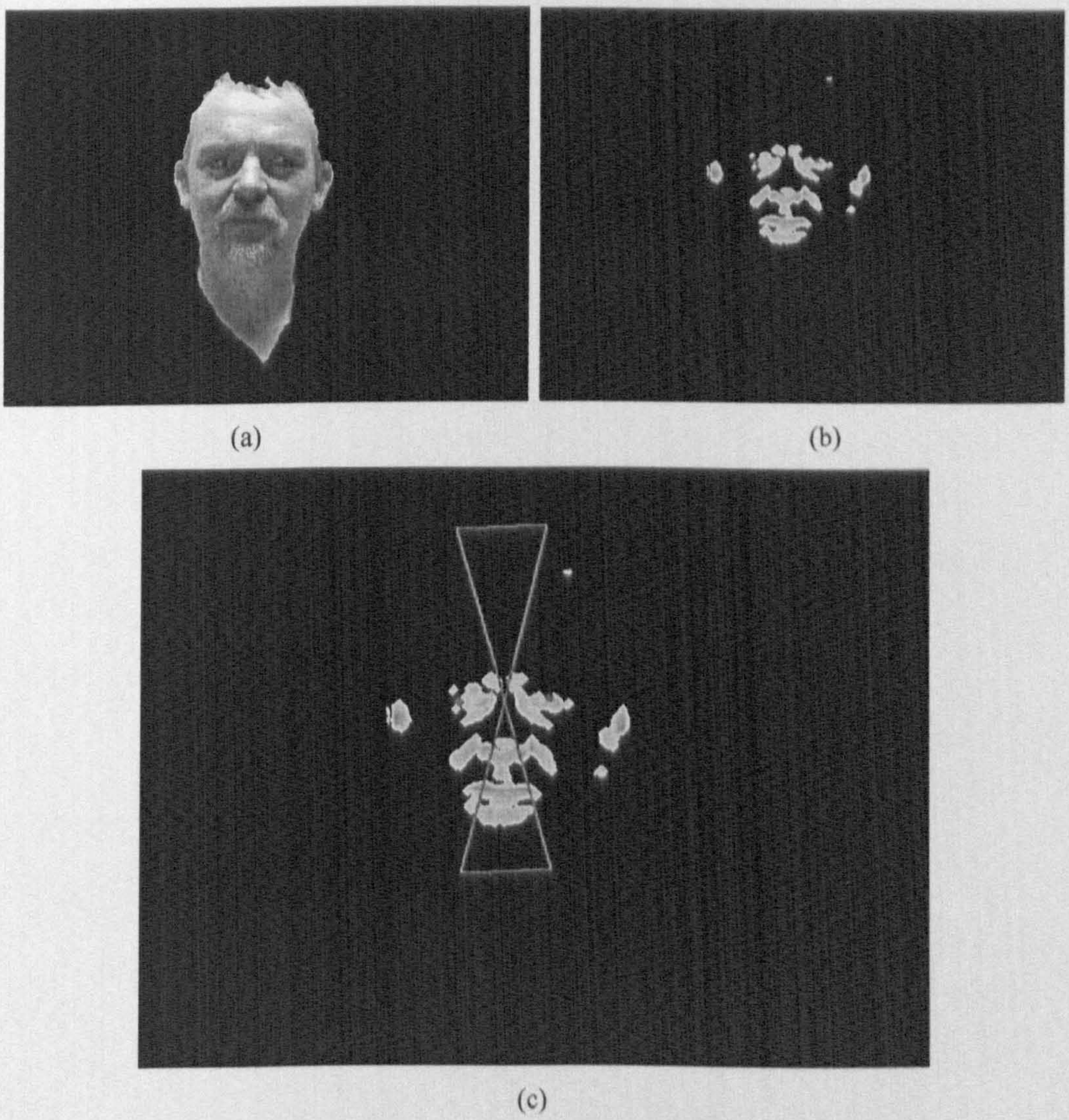
Fig 4.15 is used to illustrate the rule for verification. The two FPs at the moustache would be eliminated because there is no mouth area detected in the formed verification area. For the eye on the right, two candidates are detected. The pair of these two candidates will not pass the verification either. Yet together with the candidate of the eye on the left, they both have mouth area falling inside the verification area. In this case, rule (1) is needed. So that the pair that has larger verification area wins.

To determine if two triangles are overlapping, the Heron's formula is to be used.

$$s = \frac{a+b+c}{2}$$
$$S_{\Delta} = \sqrt{s(s-a)(s-b)(s-c)} \tag{4.6}$$

Here a, b, c represents the length of the three sides of a triangle, and $S_\Delta$ is the area of the triangle. By the calculation of equation set (4.5), we already have the coordinates of all the vertices of the verification area. By Heron's formula (4.6), we can judge if a point is inside the triangle. Given point X and a triangle ΔABC, if $S_{\Delta XAB} + S_{\Delta XAC} + S_{\Delta XAC} = S_{\Delta ABC}$, then X falls inside triangle ΔABC. From this calculation, we can know that whether two verifcation triangles are overlapping or not. To accelerate the the calculation, we will first judge if the coordinate of a point falls into the rectangle form by the two eye candidates and the two vertices. If not, then the triangle judgement is not needed.

In Fig 4.16, false positives caused by the eyebrows also have some pixels fall inside the top verification area. Then rule (2) comes into place to confirm that the lower part of the verification area contains the real mouth, which means, the orientation of the face is correctly decided.



(a) (b)

(c)

*Fig 4.16* Verification Rules (a) Eye candidates (b) Mouth area detected (c) Verification areas (d) Rules are needed for verification

## 4.4 Experimental Results

### 4.4.1 Results of the Eye Detector

As compared to other supervised eye detectors [82] [94], the algorithm complexity and computation cost of our scheme is much lower. This makes the proposed scheme applicable to real-time face detection or tracking systems.

We have our eye detector tested on the AR database [79] and Champion database [77]. Image set dbf1 from AR Database (117 images in 9 groups) and all the images

from Champion database are used for evaluation. The AR database has a uniform image size of 768x576. And the images in Champion database have an average size of 150x220.

Experimental results are shown in Table 4.1 and are encouraging. It shows that our scheme is very fast in the AR and Champion databases (average time 2.33 and 0.198 seconds per image respectively with un-optimized code), while retaining very high detection rates (100 and 92.38% correspondingly).

TABLE 4.1

RESULTS ON AR DATABASE AND YAHOO NEWS PHOTOS

| Scheme | AR Database | Champion |
|---|---|---|
| No. of Images | 117 | 1,267 |
| Stage 1. Harris corner detection | | |
| No. of FP | 3,337 | 20,901 |
| DR (%) | 100 | 93.45 |
| Time avg. (sec) | 1.738 | 0.152 |
| Stage 2. Gray map | | |
| No. of FP | 2,115 | 12,454 |
| DR (%) | 100 | 92.74 |
| Time avg. (sec) | 0.198 | 0.009 |
| Stage 3. Edge map | | |
| No. of FP | 146 | 2,443 |
| DR (%) | 100 | 92.38 |
| Time avg. (sec) | 0.394 | 0.037 |
| Final FP avg. | 1.247 | 1.93 |
| Total time avg. (sec) | 2.330 | 0.198 |

Experimental results on AR database [79] and Champion database [77] on a 1.7GHz processor

FP: False Positive, DR: Detection Rate

### 4.4.2 Results of the Face Verification

Continuing on Table 3.4, we compare our final result to Hsu's approach [18]. As shown in Table 4.2, our facial feature detection approach is 5.59 time faster than Hsu's. Although we are slower at the first two stages in the architecture, the final result is that our approach becomes 5.02 times faster in the end. As to the accuracy, we achieve a slightly higher detection rate in an image set that has more images. The average false positives are well controlled, which is 21% less. Till now we can claim

that our system outperforms Hsu's system.

TABLE 4.2

RESULTS ON THE CHAMPION IMAGE DATABASE (IMAGE SIZE ~150X220)

| Scheme | Hsu | Ours |
|---|---|---|
| No. of Images | 227 | 1,267 |
| Stage 3. Facial Feature Detection | | |
| No. of FP | 14 | 63 |
| DR (%) | 91.63% | 92.38% |
| Time avg. (sec) | 5.78 | 0.876 |
| Final DR (%) | 91.63 | 92.38 |
| Final FP avg. | 0.062 | 0.049 |
| Total Time avg. (sec) | 5.872 | 0.975 |

Results measured on a PC with 1.7GHz processor.
FP: False Positive, DR: Detection Rate

The reason that our scheme is superior is multifold. First of all, our facial feature detection scheme does not include the iterative morphological operations to detect the eyes. Our scheme bases on local heuristics to detect features and configurational information to verify the face. The computation cost for all these heuristics are very low when compared to the iterative process. The 5.59 times performance gap demonstrates that the combination of cues and configuration dramatically reduce the computation cost. The robustness and accuracy improvement of our scheme is expectable. Hsu et al exploits the color information for the eye and mouth detection. No other cues or configurational knowledge is utilized. Our scheme makes use of multiple cues including color, corner and edges. Also, analogous to human face perception skill, our face verification scheme make use of the configurational knowledge of human faces, this eliminates lots of false positives. The higher detection rate and the much lower FP average prove the combination of features and configurational knowledge a successful approach.

In Table 4.3, we demonstrate our face verification approach on the AR database. For the AR database, since our face detector now depends on the mouth area for face

verification, 3 mouth-occluded images in each series would not be able to pass the verification. So that they are not included into the test set. The result is inspiring, our face verification approach work perfectly on the AR database, leaving no false positives and achieve the 100% detection rate.

TABLE 4.3

RESULTS ON AR DATABASE AND YAHOO NEWS PHOTOS

| Scheme | AR Database |
|---|---|
| No. of Images | 90 |
| Stage 4. Face Verification | |
| No. of FP | 0 |
| DR (%) | 100 |
| Time avg. (sec) | 1.187 |
| Final FP avg. | 0 |
| Total time avg. (sec) | 5.301 |

Experimental results on AR database [79] and Champion database [77] on a 1.7GHz processor

FP: False Positive, DR: Detection Rate

## 4.5 Conclusion

In this chapter, we have proposed a fast face detector based on the skin detector of the previous chapter. The adaptive corner detection, local gray pixels and adaptive edge detection forms our eye detector. A mouth area detector using adaptive color information is also proposed. For the face verification, configurational knowledge is used. The novelty of our face detector is that we combined feature extraction and configurational information verification, which is analogous to the face perception mechanism of human beings.

Several strategies are also proposed to handle the false positives generated by the two detection processes. The experimental results on the AR database and the Champion database turn out this combination works very well. The accuracy,

performance and robustness of our scheme outperform the existing unsupervised schemes like Hsu's [18].

CHAPTER 5

# Conclusion

## 5.1 Conclusion

The main contribution of this thesis as a whole is that it showed that the combination of simple heuristics for skin/eye and mouth detection in conjunction with image processing operators achieved very good accuracy/complexity trade-offs in face detection.

In Chapter 3, the main contribution of our skin region detector is the mask refinement procedure introduced. The introduced skin region detector is proved to be very robust for real life images. Working in the normalized *RGB* color space, the proposed adaptive lighting compensation algorithm overcomes several limitations of the current lighting compensation and has fewer assumptions. A simple skin color model is set up based on much fewer samples than others and yet proved to be good enough for usage in many cases. After the skin color filter, the mask is refined using morphological operators. The morphological operators are very efficient on suppressing noises, smoothing contours and bringing back important facial features. Comparison based on performance analysis and experimental results are made to some existing algorithms.

In Chapter 4, an eye detector is first proposed. With the help of the novel adaptive corner detector, local gray pixel detection and edge map, the eye detector achieves a high detection rate with a relatively low computation cost. After the eye detection, to mimic the mechanism of human face perception, a face verification approach based on configurational information comes into place. It works by first detecting the mouth area, and then verified if it is inside the verification area of eye candidate pairs. The verification area introduced is both rotation and scale invariant. At the same time, an orientation-decision strategy and a strategy to solve the covering and overlapping problem caused by false positives from the detectors are

proposed. Also, experiments run on several face databases demonstrate the accuracy, efficiency and robustness of our scheme, when compared to existing algorithms with similar architecture.

## 5.2 Future Development

We have to admit there are still limitations in our scheme. For example, the face verification part is now limited to the cases that the mouth is not occluded. Also, there are a lot of improvements that could be done to make it go further. For the verification area, exploring the heuristics for verifying the nose in it may be able to provide a solution to the occlusion by scarf problem in the AR database [79].

On the other hand, the current implementation of our system is for the theoretical research and algorithm evaluation, so that it is not optimized for any hardware platform or specific applications. For example, our face verification algorithm is in some way similar to the problem of geometry instancing and occlusion culling in 3D computer graphic area. So that if needed, GPGPU (General-Purpose Graphic Processing Unit) programming [107] can be used help to accelerate the process. Another example of this type of optimization can be the color space transformation and our lighting compensation algorithm, GPUs can provide hardware acceleration to matrix normalization and clamping. At the same time, since the processing of our algorithm is patch based, it is highly parallelizable. Parallelism would be able to push the performance of our algorithm even higher.

For existing learning algorithms like boosting, our scheme can also be helpful. The algorithm developed and heuristics discovered in our scheme, can be used to either alleviate the computation cost, or improve the accuracy. For example, the lighting compensation algorithm can also be used to increase the constrast of the features like shape, texture or configuration extracted. The more accurate the weak classifiers are, the faster the boosting algorithm will converge and the better detection rate is to be expected. Also, our verification process can be combined into the learning methods to further suppress the false positives.

Last but not least, although our system is designed for still images, it can be used

on videos as well. For any detection or tracking system on videos, it must start with the first frame. The algorithms and heuristics of our system can be used on this first frame to extract the possible face candidates, then these candidates can be strengthen or discarded using the information in upcoming frames. For the tracking algorithms, the more accurate the first localization of the objects after, the better the tracking algorithm can perform.

# Appendix

## A.1 Color Spaces

The Color Model is defined in [108] as: An abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. Adding a certain mapping function between the color model and a certain reference color space results in a definite "footprint" within the reference color space. This "footprint" is known as a gamut, and, in combination with the color model, defines a new color space.

### A.1.1 RGB

The RGB color model is an additive model in which red, green and blue (often used in additive light models) are combined in various ways to reproduce other colors (Fig. A.1.). The RGB color space is implemented in many different ways. The most common used is the 24-bit implementation, with 8 bits, or 256 discrete levels of color per channel.

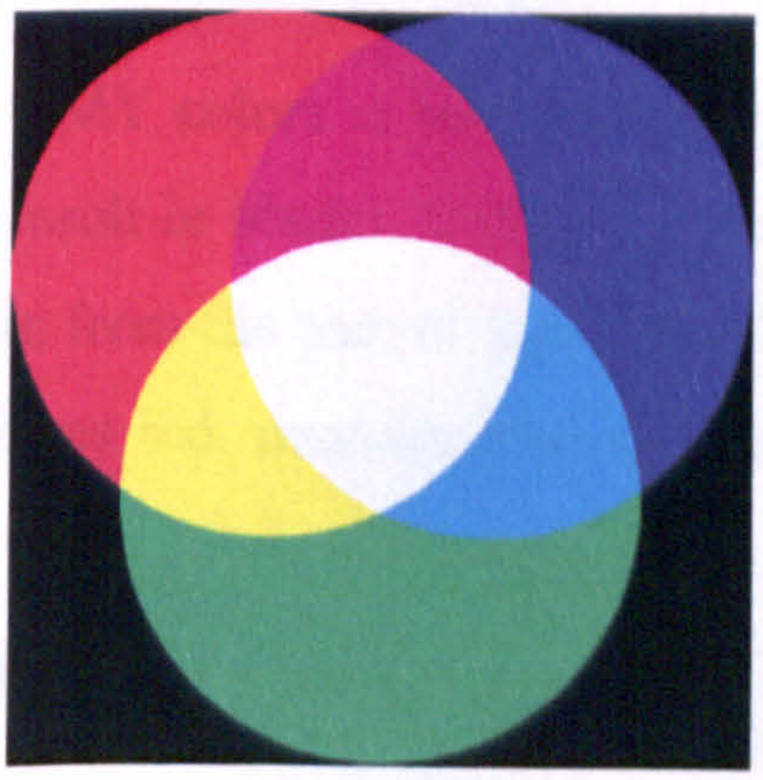


*Fig. A.1* RGB color model

Common color spaces based on the *RGB* model include *sRGB*, *Adobe RGB* and *Adobe Wide Gamut RGB*.

### A.1.2 CMYK

CMYK is a subtractive color model used in color printing. This color model is based on mixing pigments of the following colors in order to make other colors:

- C=cyan
- M=magenta
- Y=yellow
- K=key (black).

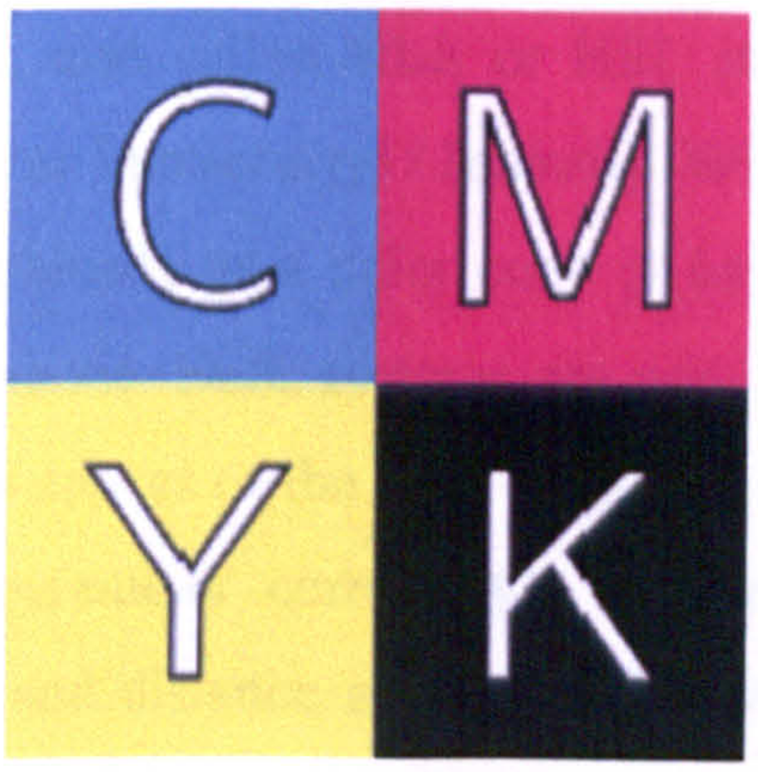



*Fig. A.2* CMYK color model

The mixture of ideal CMY colors is subtractive (cyan, magenta, and yellow printed together on white result in black). CMYK works through light absorption. The colors that are seen are from the part of light that is not absorbed. In CMYK, magenta plus yellow produces red, magenta plus cyan makes blue and cyan plus yellow generates green.

### A.1.3 YIQ/YUV/$YC_bC_r$

YIQ is a color space, formerly used in the NTSC television standard. I stands for in-phase, while Q stands for quadrature, referring to the components used in quadrature amplitude modulation. NTSC now uses the YUV color space, which is also used by other systems such as PAL.

The Y component represents the luma information, and is the only component used by black-and-white television receivers. Y comes from the standard CIE 1931 XYZ. I and Q represent the chrominance information. In YUV, the U and V components can be thought of as X and Y coordinates within the colorspace. I and Q can be thought of as a second pair of axes on the same graph, rotated 33? therefore IQ and UV represent different coordinate systems on the same plane.

### A.1.4 HSL

The HSL color space, also called HLS or HSI, stands for Hue, Saturation, Lightness (also Luminance or Luminosity) / Intensity. While HSV (Hue, Saturation, Value) can be viewed graphically as a color cone or hexcone, HSL is drawn as a double cone or double hexcone. Both systems are non-linear deformations of the RGB colour cube. The two apexes of the HSL double hexcone correspond to black and white. The angular parameter corresponds to hue, distance from the axis corresponds to saturation, and distance along the black-white axis corresponds to lightness.

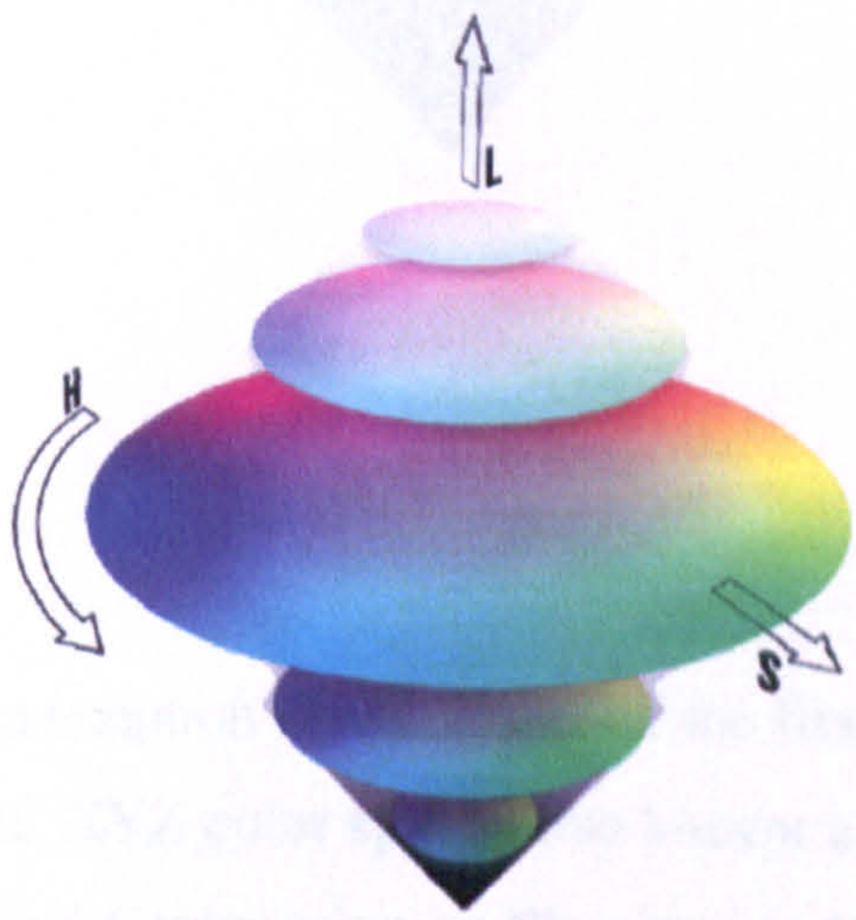



*Fig. A.3* HSV color model

### A.1.5 HSV/HSB

The HSV (Hue, Saturation, Value) model, also known as HSB (Hue, Saturation, Brightness), defines a color space in terms of three constituent components:

- **Hue,** the color type (such as red, blue, or yellow): Ranges from 0-360 (but normalized to 0-100% in some applications)
- **Saturation**, the "vibrancy" of the color: Ranges from 0-100%. Also sometimes called the "purity" by analogy to the colorimetric quantities excitation purity and colorimetric purity. The lower the saturation of a color, the more "grayness" is present and the more faded the color will appear, thus useful to define desaturation as the qualitative inverse of saturation.
- **Value**, the brightness of the color: Ranges from 0-100%

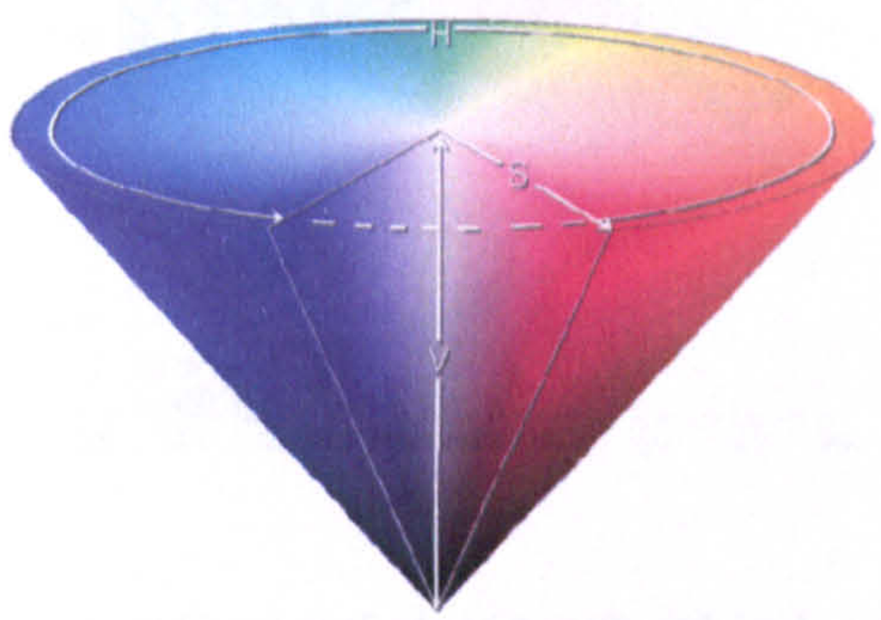



*Fig. A.4* HSV color model

### A.1.6 CIE XYZ

In the study of the perception of color, one of the first mathematically defined color spaces was the CIE XYZ color space (also known as CIE 1931 color space), created by the International Commission on Illumination (CIE) in 1931. The human eye has receptors for short (S), middle (M), and long (L) wavelengths, also known as blue, green, and red receptors. That means that one, in principle, needs three parameters to describe a color sensation. A specific method for associating three numbers (or tristimulus values) with each color is called a color space, of which the

CIE XYZ color space is one of many such spaces.

In the CIE XYZ color space, the tristimulus values are not the S, M, and L stimuli of the human eye, but rather a set of tristimulus values called X, Y, and Z, which are also roughly red, green and blue, respectively. Two light sources may be made up of different mixtures of various colors, and yet have the same color (metamerism). If two light sources have the same apparent color, then they will have the same tristimulus values, no matter what different mixtures of light were used to produce them.

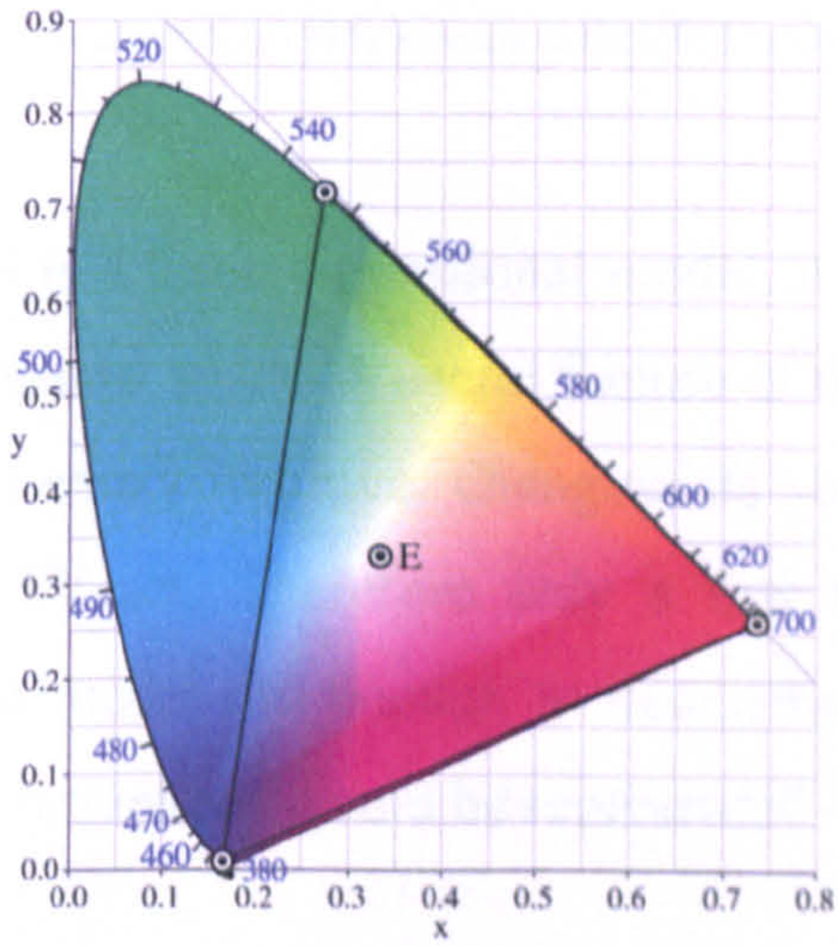


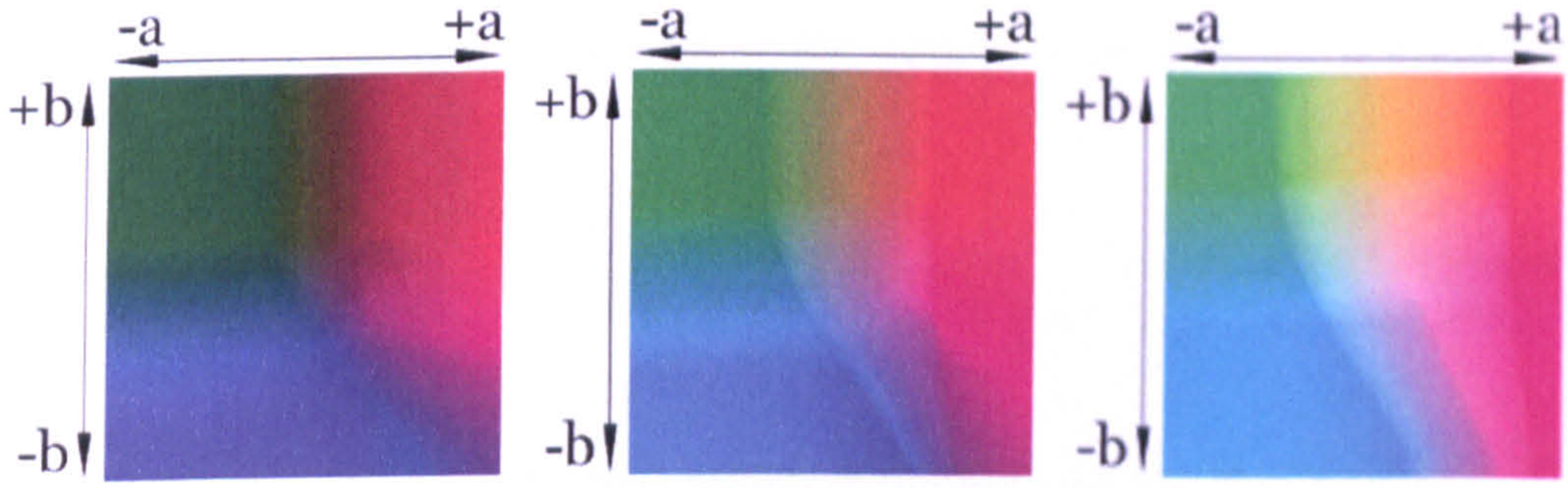

*Fig. A.5* Gamut of the CIE RGB primaries and location of primaries on the CIE 1931 *xy* chromaticity diagram.

## A.1.7 CIE L*a*b*

CIE L*a*b* (CIELAB) is the most complete color model used conventionally to describe all the colors visible to the human eye. It was developed for this specific purpose by the International Commission on Illumination (Commission Internationale d'Eclairage, hence its CIE initialism). The * after L, a and b are part of the full name, since they represent L*, a* and b*, derived from L, a and b.

The three parameters in the model represent the lightness of the color (L*, L*=0 yields black and L*=100 indicates white), its position between magenta and green (a*, negative values indicate green while positive values indicate magenta) and its position between yellow and blue (b*, negative values indicate blue and positive

values indicate yellow).




(a) (b) (c)

*Fig. A.6.* a*b* gamut at (a) 25% (b) Neutral – 50% (c) 75% lightness in the CIE L*a*b* color space

Since the Lab model is a three dimensional model, it can only be represented properly in a three dimensional space. A useful feature of the model however is that the first parameter is extremely intuitive: changing its value is like changing the brightness setting in a TV set. Therefore only a few representations of some horizontal "slices" in the model are enough to conceptually visualize the whole gamut, assuming that the luminance would be represented on the vertical axis.

## A.2 Face Databases

In the thesis, four major face databases containing color images are used to evaluate the performance and accuracy of our scheme.

### A.2.1 AR Database

The AR Database [79] is a set of images designed for use in expression analysis and face recognition. The images in the database are taken under controlled environment, with simple background. The reason that we use this database is as following. First of all, the images are of high resolution (768x576) and rich of details. It is suitable for testing the performance and scalability of a face detection algorithm.

Secondly, in each subset of the database, there are various illuminations, facial expressions and occlusions. These factors make the database useful for evaluating the robustness of the algorithms. A set of the images of the database is given in Fig. 3.24.

### A.2.2 HHI Database

This database is part of the HHI MPEG-7 content set [76]. The characteristics of the database include face orientation, skin color like background and facial expressions. The face orientation is especially useful for classifier designs.

### A.2.3 Yahoo News Database

This image database [78] is collected from the Yahoo News web site, and was first used by Hsu [18] to demonstrate the robustness of their face detector. The most challenging part of this database is that images in it are from real life. That means there will be complex background, multiple illumination, various pose and expressions, and more than one people in a single picture. All these make it the most difficult database for face detection.

### A.2.4 Champion Database

The Champion database [77] are portrait images of high school students in Nebraska, which is also introduced by Hsu [18]. The size of the images are of the smallest size in the four databases we used. The images in it have people of different sex, race, and various orientation and poses. Also the illumination is not controlled. What is more, it has the largest number of images of all. So that it is still valuable for the evaluation of both performance and accuracy of our system.

# References


[1] Nelson, C.A. The development and neural bases of face recognition. *Infant and Child Development*, 10, 3-18, 2001.

[2] Roark, D., Barrett, S.E., Spence, M.D., Abdi, H., and O'Toole, A.J. Psychological and neural perspectives on the role of facial motion in face recognition. *Behavioral and Cognitive Neuroscience Reviews*. 2(1), 15-46, 2003

[3] Bruce, V. & Young, A. Understanding face recognition. *The British Journal of Psychology*, 77 (3), 305-327, 1986

[4] J.C. Bartlett and J. Searcy. Inversion and Configuration of Faces. *Cognitive Psychology*, 25, pp. 281-316, 1993

[5] J.W. Tanaka and M.J.Farah. Parts and Wholes in Face Recognition. *Quarterly Journal of Psychology*, 46A:(2), pp. 222-245, 1993

[6] R.K. Yin. Looking at Upside-down Faces. *Journal of Experimental Psychology*, 81, pp.141-145,1969

[7] A.W. Young, D. Hellawell and D.C. Hay. Configurational Information in Face Perception, *Perception*, 16, pp.269-291, 2000

[8] Thompson, P. "Margaret Thatcher: a new illusion." *Perception*. 9(4):483-484, 1980

[9] A.J. O'Toole, D. Roark and H. Abdi. Recognition of moving faces: a psychological and neural perspective. *Trends in Cognitive Sciences*. 6, pp.261-266, 2002

[10] Stan Z.Li, A.K.Jain et al, "Handbook of Face Recognition", *Springer-Verlag*, 2005

[11] Thomas Huang, Z. Xiong, and Z. Zhang. Face Recognition Applications. *Handbook of Face Recognition*, Chapter 16, 2005

[12] M.H. Yang, D.J. Kriegman and N. Ahuja, "Detecting faces in images: A Survey", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, Jan 2002

[13] 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, Jan. 1998

[14] K. K. Sung and T. Poggio. Example-based learning for View-Based Human Face Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp.39-51

[15] A.K. Jain, R.P.W. Duin and J.Mao, "Statistical Pattern Recognition: A Review", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.22, no.1, pp.4-37, Jan 2000

[16] I. Craw, D. Tock, and A. Bennett. "Finding Face Features", *Proc. Second European Conf. Computer Vision*, pp. 92-96, 1992

[17] A. Lanitis, C.J. Taylor, and T.F. Cootes. "An Automatic Face Identification System Using Flexible Appearance Models". *Image and Vision Computing*, vol. 13, no. 5, pp. 393-401, 1995

[18] R. L. Hsu, M.Abodel-Mottaleb, A. K. Jain. "Face Detection In Color Images". *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696-706, May 2002.

[19] P.V.C. Hough, Machine Analysis of Bubble Chamber Pictures, *International Conference on High Energy Accelerators and Instrumentation*, CERN, 1959.

[20] Grecos, C. and Edirisinghe, E.A., "Two Low Cost Algorithms for Improved Diagonal Edge Detection in JPEG-LS" , *IEEE Transactions on Consumer Electronics*, vol. 3(3) , pp. 466-473, 2001

[21] M. Grudin. "On Internal Representation in Face Recognition Systems." *Pattern Recognition*, vol. 33, pp. 1161-1177, 2000.

[22] K. M. Lam and H. Yan, An Analytic-to-Holistic "Approach for Face Recognition Based on a Single Frontal View", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 771-779, 1996

[23] G. Yang and T.S. Huang. Human face detection in complex background. Pattern Recogniton, vol. 21, no. 1, pp. 53-63, 1994

[24] T.K. Leung, M.C. Burl, and P. Perona, "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching," *Proc. Fifth IEEE Int'l Conf. Computer Vision*, pp. 637-644, 1995.

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

[26] 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.

[27] S. McKenna, S. Gong, and Y. Raja, "Modelling Facial Colour and Identity with Gaussian Mixtures," *Pattern Recognition*, vol. 31, no. 12, pp. 1883-1892, 1998.

[28] J. Yang and A. Waibel, "A Real-Time Face Tracker," *Proc. Third Workshop Applications of Computer Vision*, pp. 142-147, 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] I. Craw, D. Tock, and A. Bennett, "Finding Face Features," *Proc. Second European Conf. Computer Vision*, pp. 92-96, 1992.

[31] A. Lanitis, C.J. Taylor, and T.F. Cootes, "An Automatic Face Identification System Using Flexible Appearance Models," *Image and Vision Computing*, vol. 13, no. 5, pp. 393-401, 1995.

[32] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[33] K.-K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39-51, Jan. 1998.

[34] 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, Jan. 1998.

[35] 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.

[36] H. Schneiderman and T. Kanade, "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 45-51, 1998.

[37] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri, "Finding Faces in Photographs," *Proc. Sixth IEEE Int'l Conf. Computer Vision*, pp. 640-645, 1998.

[38] M.S. Lew, "Information Theoretic View-Based and Modular Face Detection," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pp. 198-203, 1996.

[39] A.J. Colmenarez and T.S. Huang, "Face Detection with Information-Based Maximum Discrimination," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 782-787, 1997

[40] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.

[41] A.M. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, Feb. 2001.

[42] E.J. Lee and Y.H Ma. Automatic flesh tone reappearance of color enhancement in TV. *IEEE Trans on Consumer Electronics*, 43(4), pp.1153-1159, 1997

[43] M. Abdel-Mottab and A. Elgammal. "Face Detection in Complex Environments from Color Images", *IEEE Int'l Conf. Image Processing*, pp. 622-626, Oct 1993.

[44] H. Wu, Q. Chen, and M. Yachida. "Face Detection from Color Images Using a Fuzzy Pattern Matching Method", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 557-563, Jun 1999.

[45] Garcia C. , Tziritas G, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis", *IEEE Trans. on Multimedia*, 1(3), p.264-277, Sep 1999

[46] G. Wyszecki and W.S. Stiles. Color Science Concepts and Methods, Quantitative Data and Formulae, Second Edition. *Wiley, New York*, 2000

[47] J. Brand and J.S. Mason. A comparative assessment of the three approaches to pixel-level human skin detection. *Proceedings of International Conference on Pattern Recognition*. Barcelona. 1:5056-5059, 2000

[48] T.S Caetano, S.D. Olabarriaga, and D.A.C Barone. *Do mixture models in chromaticity space improve skin detection?* Pattern Recognition, 36(12), pp.3019-3021, 2003

[49] B. Martinkauppi. Face Colour under Varying Illumination – Analysis and Applications. *PhD thesis, University of Oulu*, 2002

[50] B. Funt, K. Bernard, and L. Martin. "Is Machine Color Constancy Good Enough". *Proceedings of 5th European Conference on Computer Vision (ECCV'98)*, pp. 445-449, Freiburg, Germany, 1998

[51] M.J.Jones and J.M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1), pp.81-96, 2002

[52] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. "Color by Correlation: A Simple, Unifying Framework for Color Constancy". *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209-1221, 2001

[53] J. C. Terrillon, M. N. Shirazi, H. Fukamachi, S. Akamatsu. "Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images", *Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 54-61, 2000.

[54] F. Solina, P. Peer, B. Batagelj, and S. Juvan, "15 Seconds of Fame – An Interactive, Computer-Vision Based Art Installation", *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARCV 2002)*, pp. 198-204, Singapore, 2002.

[55] M. C. Shin, K. I. Chang and L. V. Tsap, "Does Colorspace Transformation Make Any Difference on Skin Detection?", *IEEE Workshop on Applications of Computer Vision*, Orlando, FL, Dec 2002

[56] Zarit, Super, and Quek. "Comparison of five color models in skin pixel classification". *ICCV'99 Int'lWorkshop on recognition, analysis and tracking of faces and gestures in Real-Time systems*, 1999.

[57] M.H. Yang and N. Ahuja. "Face Detection and Gesture Recognition for Human-Computer Interaction." *Kluwer Academic*, New York, 2001

[58] L.M. Bergasa, M. Mazo, A. Gardel, M.A. Sotelo, and L. Boquete. Unsupervised and adaptive Gaussian skin-color model. *Image and Vision Computing*, vol. 18, no. 12, pp. 987-1003, 2000

[59] T. Piirainen, O. Silven and V. Tuulos. Layered self-organizing maps based video content classification. In *Proceedings of Workshop on Real-time Image Sequence Analysis*, Oulu, Finland, pp. 89-98, 2000

[60] L.M. Son, D. Chai, and A. Bouzerdoum. A universal and robust human skin color model using neural networks. *In Proceedings of International Joint Conference on Neural Networks*, Washington DC, vol. 4, pp. 2844-2849, 2001

[61] B. Schiele and A. Waibel. Gaze tracking based on face-color. In *Proceedings of International Workshop on Automatic Face and Gesture-Recognition*, Zurich, pp. 344-348, 1995

[62] D. Chai and K.N. Ngan. Locating facial region of a head-and-shoulders color image. In Proceeeding of 3rd International Conference on Automatic Face and Gesture Recognition, Nara, Japan, pp. 124-129, 1998

[63] B. Martinkauppi, M. Soriano, and M. Pietikainen. Detection of skin color under changing illumination: a comparative study. In *Proceedings of 12th International Conference on Image Analysis and Processing*, Mantova, Italy, pp. 652-657, 2003

[64] B. Martinkauppi, P. Sangi, M. Soriano, M. Pietikainen, S. Huovinen and M. Laaksonnen. Illumination invariant face traking with mean shift and skin locus. In *Proceedings of IEEE International Workshop on Cues in Communication*, Kauai, Hawaii, pp. 44-49, 2001

[65] M. Storring, H.J. Andersen, and E. Granum. Physics-based modeling of human skin color under mixed illuminants. Journal of Robotics and Autonomous Systems, vol. 35(3-4), pp. 131-142, 2001

[66] R.L. Hsu. Face Detection and Modeling for Recogntion. *PhD thesis.*, Michigan State University, 2002

[67] T. Ohtsuki and G. Healy. Using color and geometric models for extracting facial features. *Journal of Imaging Science and Technology*, vol. 42, no. 6, pp. 554-561, 1998

[68] P.Soille. "Morphological Image Analysis: Principles and Applications, Second Edition". Springer-Verlag, pp. 96-97, 2002.

[69] van den Boomgard, Rein, and Richard van Balen, "Methods for Fast Morphological Image Transforms Using Bitmapped Images," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, Vol. 54, No. 3, pp. 252-254, May 1992.

[70] Bovik and D.Desai. Basic binary image processing. Handbook of Image and Video Processing, *Academic*, pp. 37-53, 2000

[71] Sandeep and Rajagopalan. "Human Face Detection in Cluttered Color Images Using Skin Color and Edge Information". 2002

[72] A. K. Jain et al, "Data Clustering: A Review", *ACM computing surveys*, 1999

[73] M.Laddes, J.C.Vorbruggen, J.Buchman,J.Lange, C.V.Malsburg, R.P.Wurtz and W.Konen, "Distortion Invariant object recognition in the Dynamic Link Architecture", *IEEE Trans. On Computers*, 42(3):300-311, 1993.

[74] L.Wiskott,J.Fellous,N.Kruger, C.V.Malsburg, Face Recognition by Elastic Graph Bunch Matching, *Intelligent Biometric Techniques in Fingerprint and Face Recignition, L.C.Jain and al.(eds)*,Springer-Verlag 1999

[75] D.Chai, K.N.Ngan, "Face Segmentation Using Skin-Color Map in Videophone Applications", *IEEE Trans. Circuits and Systems for Video Technology*, vol.9, no.4, pp. 551-564, 1999

[76] MPEG7 Content Set from Heinrich Hertz Institute, http://www.romsquared.com/mpeg7.htm

[77] Champion database, http://www.libfind.unl.edu/alumni/events/breakfast_for_champions.htm

[78] Yahoo news photos, http://www.cse.msu.edu/~hsureinl/facloc/index_facloc.db.html

[79] A.M. Martinez and R. Benavente. The AR Face Database. *CVC Technical Report* #24, June 1998

[80] P.J. Phillips, Hyeonjoon Moon, S.A. Rizvi, and P.J. Rauss, The feret evaluation methodology for face-recognition algorithms, no. 10, 1090–1104., *IEEE Transactions on PAMI*, 2000

[81] A.M. Martinez, Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class, no. 6, 748–763, *IEEE Transactions on PAMI*, 2002.

[82] P.Wang, M.B.Green, Q.Ji and J.Wayman, Automatic Eye Detection and Its Validation, *IEEE Workshop on Face Recognition Grand Challenge Experiments (with CVPR)*, San Diego, CA, June 2005

[83] Qiang Ji, Harry Wechsler, Andrew Duchowski, and Myron Flickner, Special issue: eye detection and tracking, 1–3, *Computer Vision and Image Understanding*, 2005.

[84] Zhiwei Zhu, Qiang Ji, and Kikuo Fujimura, Combining kalman filtering and mean shift for real time eye tracking under active IR illumination, pp. 318–321, *International Conference on Pattern Recognition*, 2002.

[85] J. Huang, H. Wechsler. Visual routines for eye location using learning and evolution. 73-82, *IEEE Transactions on Evolutionary Computation*, 2000.

[86] A. Pentland, B. Moghaddam, T. Starner. View-based and modular eigenspaces for face recognition. In P*roceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp.84-9, Seattle, WA, 1994.

[87] F. S. Samaria, A. C. Harter. Parameterization of a stochastic model for human face identification. In *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, pp.138-142, Sarasota, FL, 1994.

[88] R. Kothari and J.L. Mitchell, Detection of eye locations in unconstrained visual images, In *Proceedings of ICIP*, vol. 3, pp. 519–522, 1996.

[89] Z.H. Zhou and X. Geng, Projection functions for eye detection, *Pattern Recognition*, no. 5, 1049–1056, 2004.

[90] K. M. Lam, H. Yan. Locating and extracting the eye in human face images. *Pattern Recognition,* vol. 29, no.5, pp. 771-779, 1996.

[91] T. d'Orazio, M. Leo, G. Cicirelli, and A. Distante, An algorithm for real time eye detection in face images, ICPR, pp. 278–281, 2004.

[92] L. Chen and C. Grecos, A Fast Skin Color Detector for Face Extraction, In *Proceedings SPIE Electronic Imaging Conference*, 2005

[93] C.G. Harris and M.J. Stephens. A combined corner and edge detector, In *Proceedings Fourth Alvey Vision Conference*, 1988, pp. 147-151.

[94] C.Morimoto, D.Koons, A.Amir and M.Flickner, Real-Time Detection of Eyes and Faces, *PUI Workshop*, 1998

[95] S. Kawato and N. Tetsutani. Real-time detection of Between-the-Eyes with a Circle Frequency Filter. In *Proceedings of the 5th Asian Conference on Computer Vision*, Melbourne, Australia, 2002

[96] K. Fukui and O. Yamaguchi. Facial feature point extraction method based on combination of shape extraction and pattern matching. *Systems and Computers in Japan*, 29(6):49–58, 1998.

[97] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Realtime face and facial feature tracking and applications. *Proc. Workshop on Audio-Visual Speech Processing*, pp. 79–84, 1998.

[98] K. Mikolajczyk and C. Schmid, "Indexing Based on Scale Invariant Interest Points," *Proc. Eighth Int'l Conf. Computer Vision*, pp. 525-531, 2001.

[99] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," *Proc. Seventh European Conf. Computer Vision*, pp. 128-142, 2002.

[100] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool, "A Comparison of Affine Region Detectors," accepted by *Int'l J. Computer Vision*, 2005

[101] N.C Raghavendra, A.R. Dasu, S. Panchanathan. Complexity Analysis of Sprites in MPEG-4. In *Proc. SPIE Media Processors*, Vol. 4313, p. 69-73, 2001

[102] D. Lowe, "Object recognition from local scale-invariant features", *ICCV*, pp. 1150-1157, 1999.

[103] Yan Ke and Rahul Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors", *CVPR*, 506-503, 2004.

[104] Krystian Mikolajczyk and Cordelia Schmid, A Performance Evaluation of Local Descriptors, IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, 2005

[105] M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 12, December 1997, pp. 1338-1359.

[106] J. Canny, A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-714, 1986

[107] D. Tarditi, S. Puri and J. Oglesby. Accelerator: simplified programming of graphics processing units for general-purpose uses via data parallelism. *Microsoft Research Technical Report*, MSR-TR-2005-184, 2005

[108] Color Space. *Wikipedia, the free encyclopedia.* http://en.wikipedia.org/wiki/Color_space

# Publications

L. Chen and C. Grecos, "A Novel Algorithm for Skin-tone Color Filtering in Color Images", In *Proceedings IASTED Modeling and Simulation,* Marina del Rey, California, 2004

L. Chen and C. Grecos, "A Fast Skin Color Detector for Face Extraction", In *Proceedings SPIE Electronic Imaging Conference*, San Jose, California, 2005

L. Chen and C. Grecos, "A Face Skin Region Detector for Color Images", In *Proceedings IEE Visual Information Engineering*, Glasgow, Scotland, 2005

L. Chen and C. Grecos, "A Fast Eye Detector using Corners, Color and Edges", In *Proceedings SPIE Electronic Imaging Conference*, San Jose, California, 2006

L. Chen and C. Grecos, "Eye Detection with Multiple Cues and Skin Color Filter", Accepted by *EPSRC International Centre for Advanced Research in Identification Science*, Sheffield, 2006