Human Face Detection in Color images

by

Jun Gao

A Project
presented to Ryerson University

in partial fulfillment of the
requirements for the degree of
Master of Engineering

in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2004
Jun Gao 2004

Instructions on Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this project. Please sign below, and give address and date.

Abstract

Detection of human face has many realistic and important applications such as human and computer interface, face recognition, face image database management, security access control systems and content-based indexing video retrieval systems. In this report, a face detection scheme will be presented. The scheme is designed to operate on color images. In the first stage of algorithm, the skin color regions are detected based on the chrominance information. A color segmentation stage is then employed to make skin color regions to be divided into smaller regions which have homogenous color. Then, we use the iterative luminance segmentation to further separate the detected skin region from other skin-colored objects such as hair, clothes, and wood, based on the high variance of the luminance component in the neighborhood of edges of objects. Post-processing is applied to determine whether skin color regions fit the face constrains on density of skin, size, shape, and symmetry and contain the facial features such as eyes and mouths. Experimental results show that the algorithm is robust and is capable of detecting multiple faces in the presence of a complex background which contains the color similar to the skin tone.

Keywords: Face detection, Skin color detection, Color segmentation

Acknowledgements

I would like to thank Prof. Bobby Ngok-Wah Ma, my supervisor, for his patient support and advice.

Table of Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

Detection of human face has gained many important and realistic applications such as intelligent human computer interaction, face recognition, face image database management, video surveillance, security access control systems, image annotation systems, and content-based indexing video retrieval systems [1,2,7,9]. Usually, face detection is the first and crucial step in applications of any face processing system. For instance, in the cases of face recognition approaches and face tracking approaches, the existence and position of faces in an image and an image sequence are supposed to have been identified and localized. The aim of the face detection is to figure out whether or not there are human faces including a single face or multiple faces in the images, and if there are, mark their locations and extract their spatial extent regardless of its three-dimensional position, orientation, and lighting conditions. Although face detection is a rather simple task to the human eyes, the process of automatic detection by a computer is not straightforward because different people have different face characteristics in size, shape, color, and texture. Further, many internal and external factors can also affect the features of the faces greatly, such as pose, image orientation, lighting condition, facial expression, partial occlusion, presence or absence of structural components (beards, mustaches, glasses). Therefore, in order to build a fully automated system which can analyze the information contained in face images, it is necessary to devise a robust and efficient algorithm for the face detection.

In recent year, various approaches for the face detection on grayscale images or color images have been developed. A comprehensive survey about the recent developments on face detection can be found in [1]. Some of the most representative works include principal component analysis (PCA)[14], neural network [15], deformable template matching [16], geometrical feature modeling [3,9], skin color analysis [2,3,4,7], Hough transform [3], and fuzzy theory matching [13].

Among the various approaches of face detection, color analysis has shown a promising prospect. Although color-based approaches have difficulties in robustly detecting skin colors when the condition of background is complex and the lighting conditions is either too dark or too bright, color information is still useful in detecting face and specific facial features if a proper skin color model can be established. Besides, under the constant lighting condition, the skin color has presented the almost invariant characteristic no matter what changes have happened in the orientation, size and partial occlusion of the face. Moreover, comparing with the processing speed for other facial feature analyses, the processing speed for color information analysis is much faster. This characteristic is very important in the operations of video schemes [21]. Several color spaces are used for color analyses in skin color detection. These color spaces include RGB, normalized RGB, HSV (or HSI), YCrCb, YIQ, YES, CIE XYZ, and CIE LUY [1].

In this study, a face detection scheme has been presented. The scheme is designed to operate on color images. The color spaces we discuss are HSV color space (H and S represent chromatic components, V represents luminance component.) and YCrCb color space (Cr and Cb represent chromatic components, Y represents luminance component.). Through comparison, we found that the scheme of the face detection is easier to implement in HSV color space. In the first stage of algorithm, the skin color regions are detected based on the chrominance information and the non-skin pixels are discarded as many as possible. An unsupervised color segmentation stage is employed to make skin color regions to be divided into smaller regions which have homogenous color. Then, we use the iterative luminance segmentation to further separate the objects with similar skin color based on the high variance of the luminance component in the neighborhood of edges of objects. Finally, Post-processing is applied to determine whether skin color regions fit the face constrains on density of skin, size, shape, and symmetry and contain the facial features such as eyes and mouth.

The face detection scheme presented in this study has the following distinguishing features:

(1) Unsupervised color segmentation algorithm for face detection was presented in [21]. Iterative segmentation in the luminance component was applied for face detection in [2]. In this study, we combined the modified unsupervised color segmentation method and iterative luminance segmentation method to generate face candidate regions. This scheme could efficiently detect faces even when there are other skin-colored objects in complex background environments.

(2) In the stage of unsupervised color segmentation, instead of using the watershed base technique to cluster 2-D histogram in YCrCb color space [21], we employed the watershed base technique to cluster the 1-D histogram in HSV color space. The proposed method was found to be effective and has relatively low computation complexity.

(3) Based on the concept presented in [21], we designed a Label Flood Fill method to fill the unassigned pixels after label morphological filtering. This method not only removes all unassigned pixels effectively, but at the same time may improve the shape of the border between the two different color regions.

The organization of this project report is as follows. Chapter 2 introduces the face detection algorithm which includes skin tone detection, unsupervised color segmentation, iterative luminance segmentation, and post-processing. Chapter 3 presents experimental and observed results. Chapter 4 presents the conclusion and proposes an approach for the further enhancement of the algorithm.

Chapter 2

Face Detection Algorithm

An overview of our face detection algorithm is illustrated in Figure 2.1. Each input color image will go through five stages. The whole procedure follows a coarse-to-fine strategy.

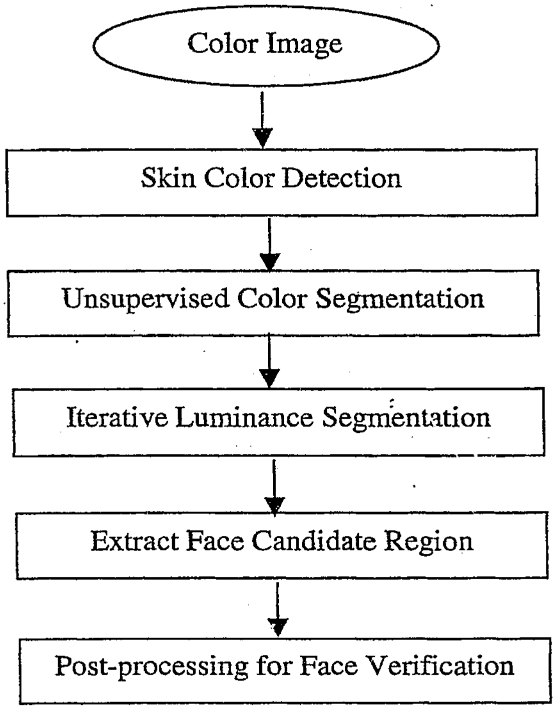

Color Image
Skin Color Detection
Unsupervised Color Segmentation
Iterative Luminance Segmentation
Extract Face Candidate Region
Post-processing for Face Verification

Figure 2.1 Face detection algorithm

2.1 The detection of skin color region

The first step of skin color region detection is to employ the skin chrominance information to locate the potential face areas and reject as much non-face parts in the image as possible. As we process the color images, one of the important issues is the selection of the color space. Although different people have different skin colors, several studies have shown the major difference among different skin colors lies in luminance or intensity rather than chrominance components [4]. Therefore, we are more interested in those color spaces in which it is possible to separate the luminance component from the chromatic components. By assuming that luminance component has little influence on the distribution of the chromatic components, we discard luminance component in skin color detection stage. The assumption is valid under the normal lighting condition. In our study, we discuss and use two color spaces, YCbCr color space and HSV (Hue, Saturation, and Value) color space for the detection of skin color regions.

(1) YCbCr color space

In YCbCr color space, the luminance information is in Y component and the chrominance information is contained in Cb and Cr. The following set of formulas is used to convert the RGB (Red, Green, Blue) components into YCbCr components.

$$\begin{aligned} Y &= 0.257*R + 0.504*G + 0.098*B + 16 \\ Cb &= 0.148*R - 0.291*G + 0.439*B + 128 \\ Cr &= 0.439*R + 0.368*G - 0.071*B + 128 \end{aligned} \tag{2.1}$$

(2) HSV color space

HSV is often chosen as color representation because it corresponds closely to the human intuition on the color. H and S components contain the chromatic information and V includes the luminance information. HSV can be best described by the cone mode (see Figure 2.2). Hue is the attribute of color and is measured by the angle around the vertical axis. It looks like a color wheel which is denoted by red, yellow, green, cyan, blue, Magenta and so on. Saturation is used to describe how pure a color is or how much white is added to a color. Value is a measure of the relative brightness. The RGB components can be converted into HSV components through the following formula [6],

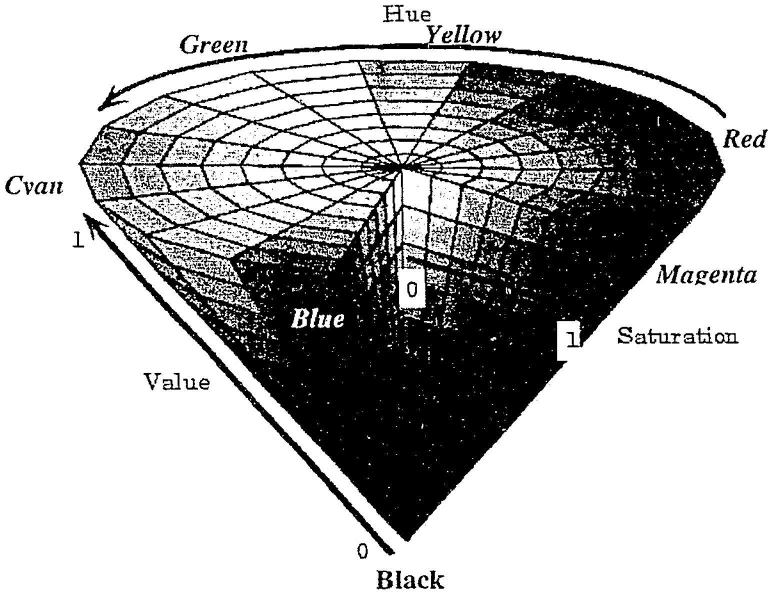


Figure 2.2 The HSV color space

$$H = \begin{cases} H_1\ if & B \le G \\ 360^0 - H_1\ if & B > G \end{cases}$$

Where

$$H_1 = \cos^{-1}\left\{\frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right\} \qquad (2.2)$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)}$$

$$V = \frac{\max(R,G,B)}{255}$$

In Figure 2.2, S and V have been normalized within the range [0,1].

We use the two color spaces to analyze 60 manually marked skin samples which cover a large range of skin color appearances with different races and different light conditions. Figure 2.3 (a) and Fig 2.3 (b) shows the distributions of Cb-Cr plane and S-H plane of these samples. We observe that the skin color samples form a small and compact cluster in Cb-Cr plane and S-H plane. The result of this observation gives a good indication on whether a pixel is the part of the skin or not. We also notice that the cluster of skin color pixels in H-S plane is less compact than cluster of skin color pixels in Cb-Cr Plane.

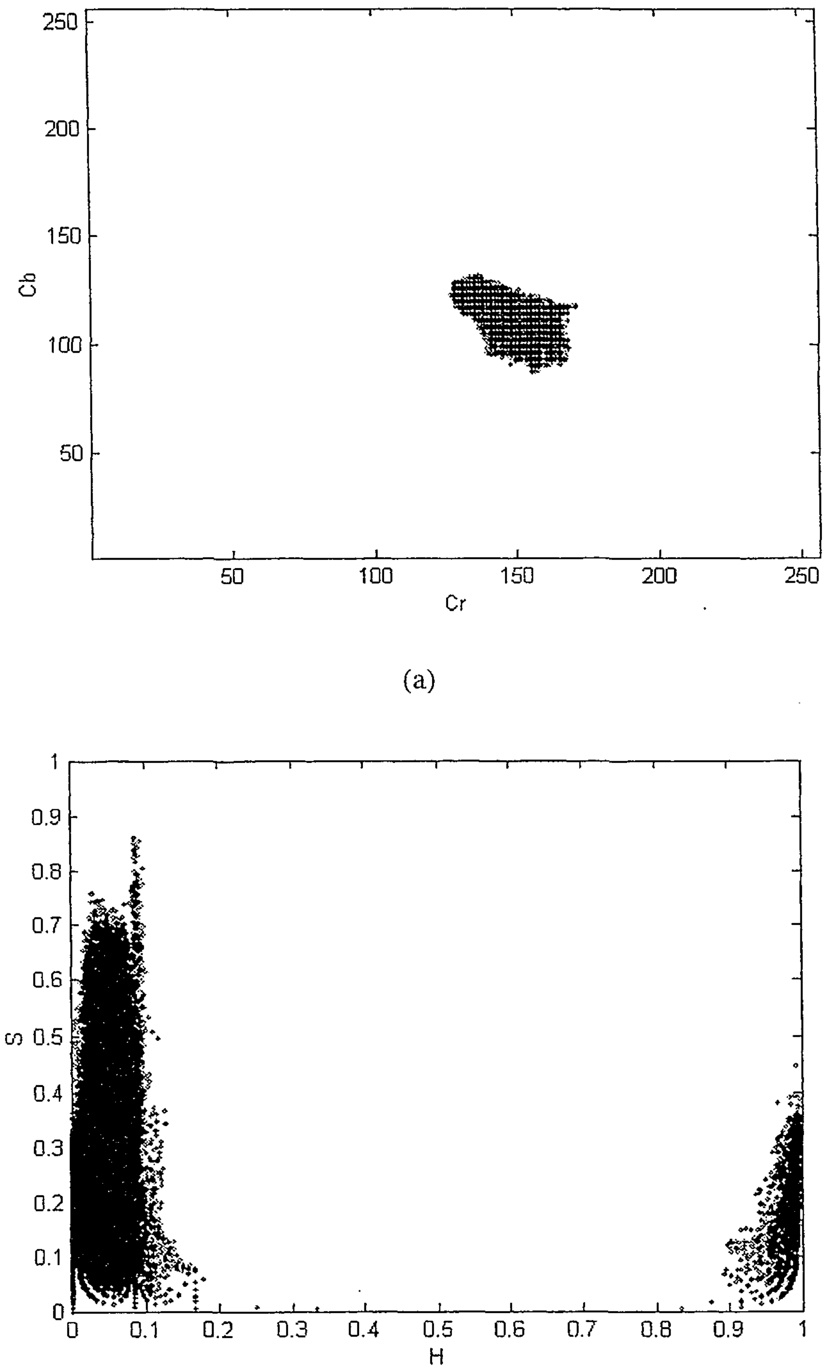
We consider the following three methods to classify whether each pixel in the image belong to skin color or non-skin color.

(1) Histogram threshold method

Histogram distribution of Cb and Cr is shown in Figure 2.4. We apply the maximum and minimum thresholds to both Cb and Cr components. If the Cb and Cr component values of the pixel are within the range of the thresholds, the pixel is classified as the skin pixel, otherwise, it is considered as non-skin pixel. If we broaden the range of the thresholds, some non-skin pixels might be detected. If we narrow down the range of the threshold, some actually skin pixels might be rejected. Based on Figure 2.4, we choose the thresholds as following:

$$130 <= Cr <= 162 \qquad (2.3)$$

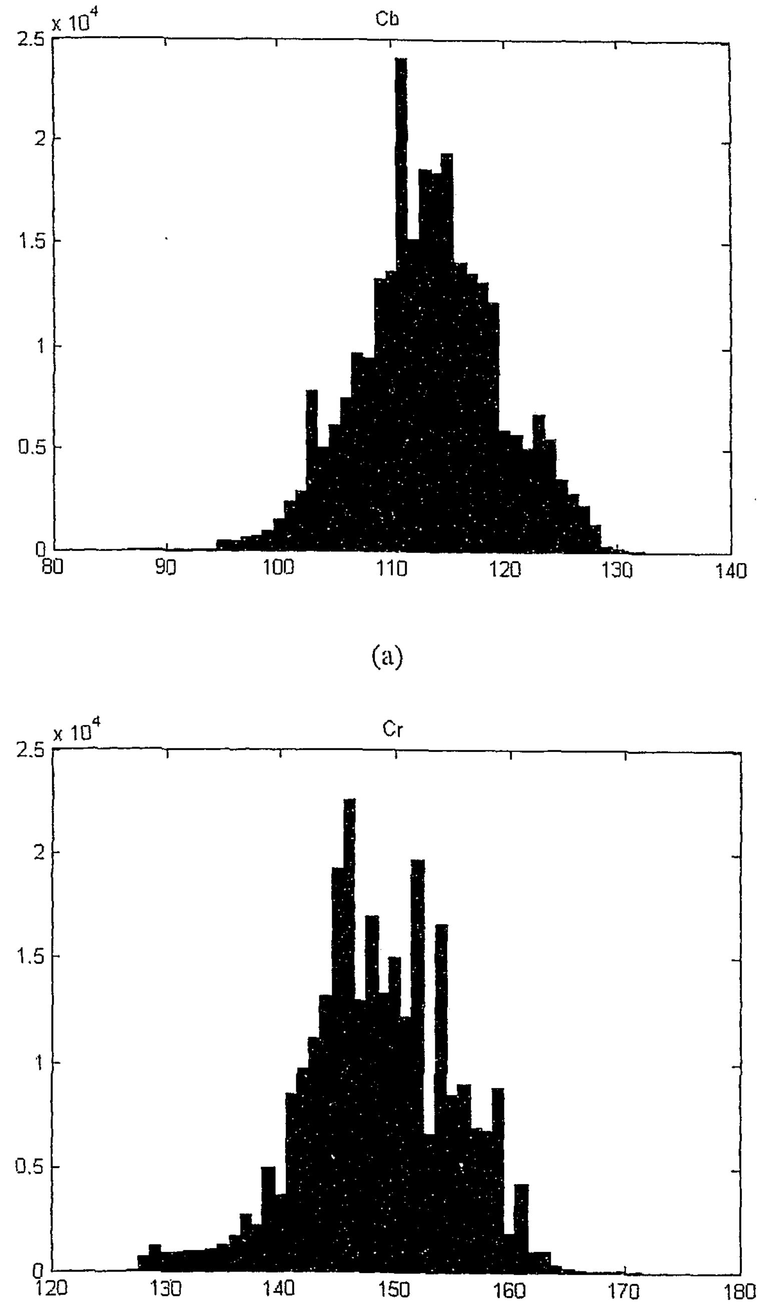
98 <= Cb <= 126



(a)

(b)

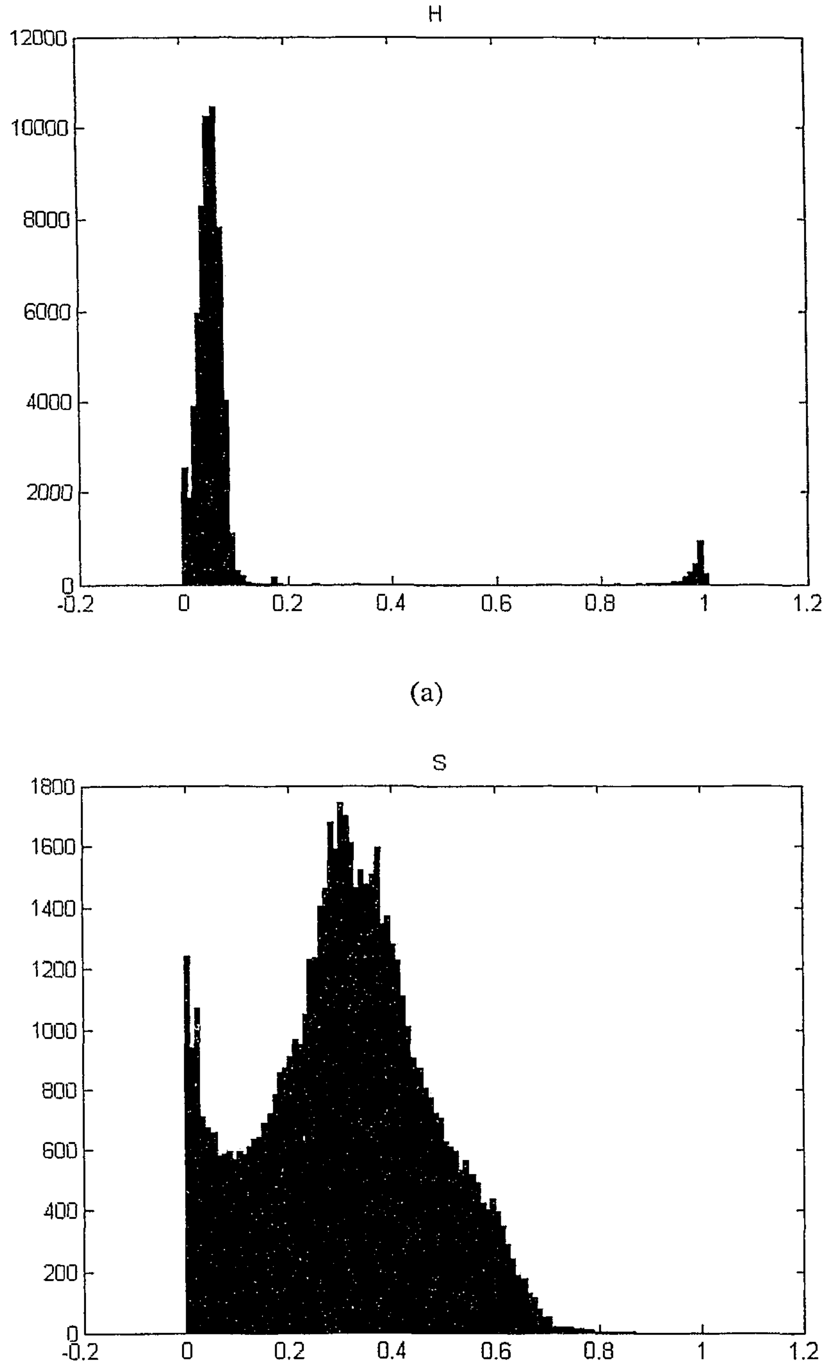
Figure 2.3 (a) Distribution of Cb and Cr (b) Distribution of H and S



(a)

(b)

Figure 2.4 Histogram distributions of Cb and Cr



(a)

(b)

Figure 2.5 Distributions of H and S

Similar to Cb-Cr plane, we can find the similar threshold for H and S distribution. As we have mentioned, the distribution of H and S, especially for S component is more scattered than the distribution of the Cb and Cr. Figure 2.5 is the histogram distributions of H and S. Furthermore, Hue is unreliable when Value and Saturation is very low. Consequently, we should set lower bound for both Value an‹ Saturation. We will further discuss this situation in the color segmentation stage.

(2) Skin color probability model method

Several researches have indicated that the distribution of the human skin can be modeled as the Gaussian distribution [4,6]. We observe that the distribution of the skin color for Cb and Cr can be expressed as a bi-variate Gaussian distribution $N(\mu,\Sigma)$ where

$$\mu = (\mu_b \quad \mu_r)^T$$
$$\Sigma = \begin{vmatrix} \delta_{bb}^2 & \delta_{br}^2 \\ \delta_{rb}^2 & \delta_{rr}^2 \end{vmatrix} \qquad (2.4)$$

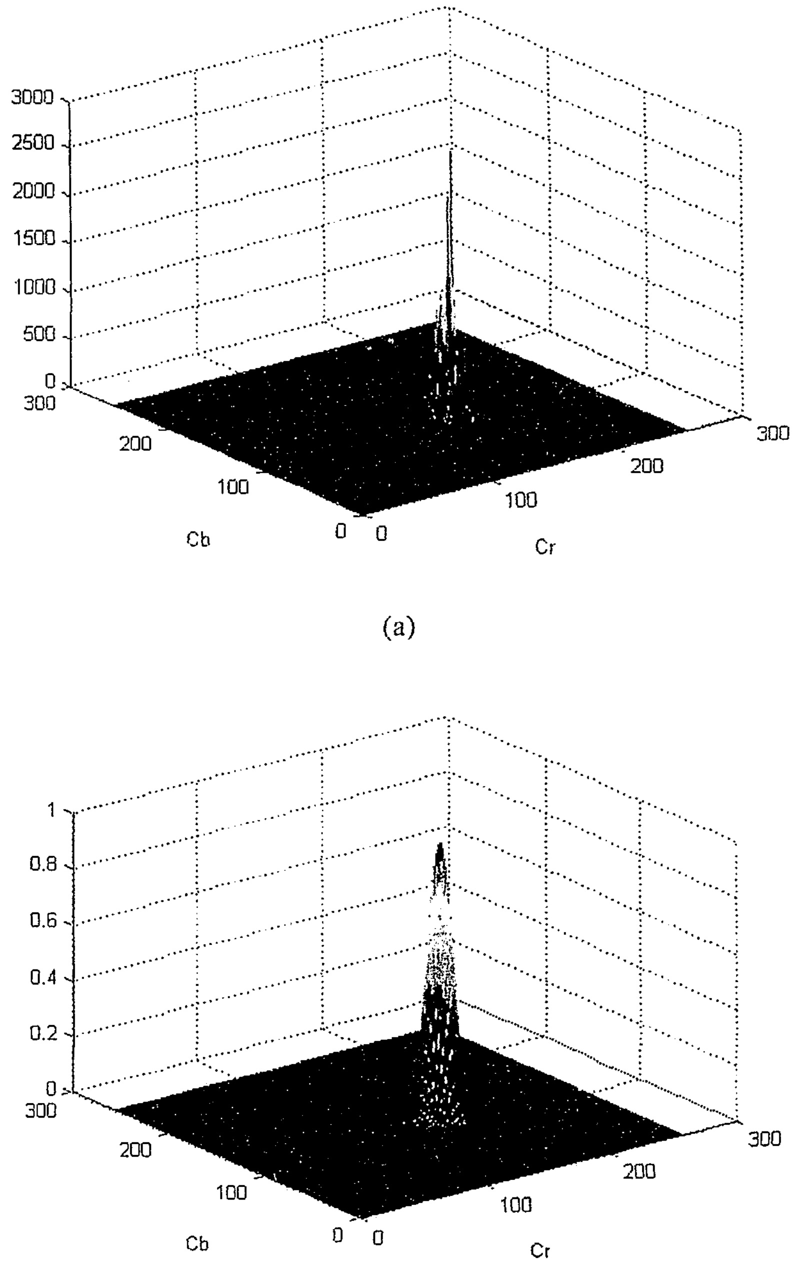
represent mean and covariance, respectively.

Figure 2.6 (a) shows distribution of skin samples for Cb and Cr. Figure 2.6 (b) shows the model of skin color for Cb and Cr based on mean and covariance of the skin samples. The probability of whether a pixel belongs to a skin region is determined by

$$P(r,b) = \exp[-0.5(X-\mu)^T \Sigma^{-1}(X-\mu)] \qquad (2.5)$$

where

$\mu = E\{X\}$ and $X = (r \quad b)^T$



(a)

(b)

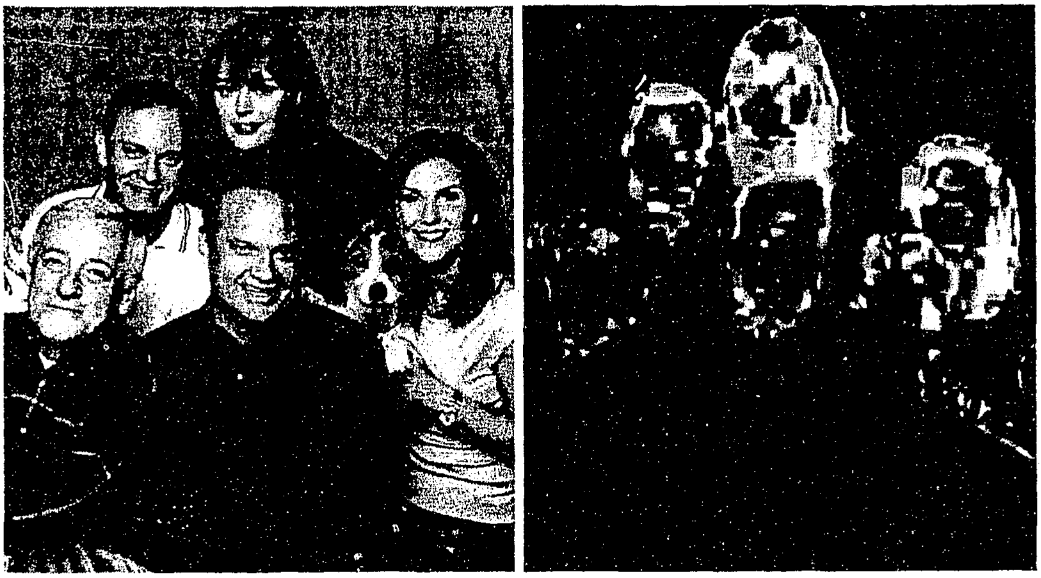
Figure 2.6 (a) Distribution of skin color for Cb, Cr

(b)The Model of skin color for Cb, Cr

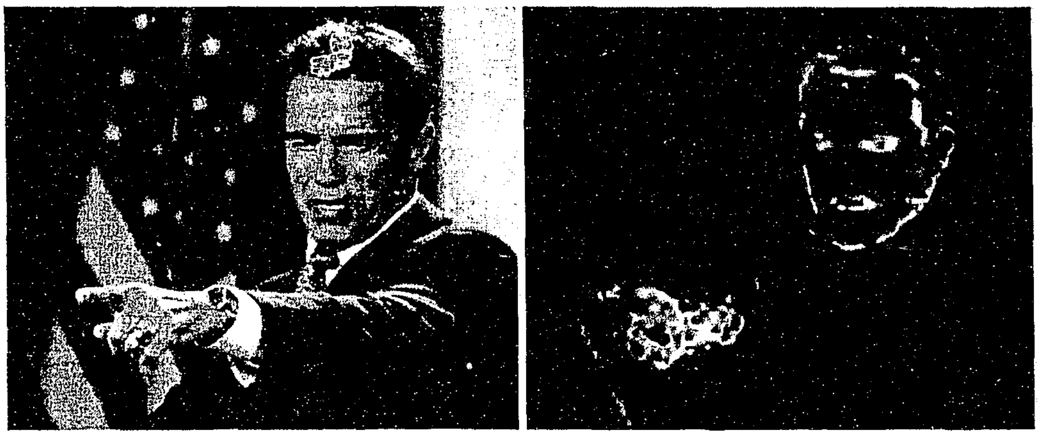
If the probability of a pixel is greater than a threshold, it is identified to have skin color. Otherwise, it is considered as non-skin color.

In practice, we found that it is very difficult to choose a proper threshold. If the skin color of an image has higher probability in the skin color model, we could obtain a better result of the skin detection. However, if the skin color of an image has a lower probability in skin color model, the method may not be able to identify the skin color in the pixels.

Figure 2.7 shows the skin color probability image and the original image. Comparing with the histogram threshold method, we think the histogram threshold method is more simple, efficient, and suitable for skin detection.



(a) (b)



(c) (d)

Figure 2.7 Original image and the skin color probability image

(3) Planar envelope approximation

In histogram threshold method, the thresholds of Cb and Cr (see formula 2.3) form a rectangular region to discriminate the skin color pixel and non-skin pixels. But the actual distribution of human skin color is an irregular shape (see Figure 2.3). Therefore, some actual skin color pixels might be excluded and some non-skin color might be included. In addition, the method assumes that luminance component has little influence on the distribution of the chromatic components. But in extreme light situation such as too dark or too bright, the distribution of chromatic components will be affected by the luminance component. A more accurate method, the planar envelop approximation for both YCbCr color space and HSV color space has been proposed in [7]. In this method, several bounding planes are found to form a 3-D envelop which is used to approximate the human skin color clusters. Through comparison, we found the planar envelope approximations have more widely adaptation for various color images. Further, the planar envelope approximation for HSV color space is easier to implement compared with the envelope approximation for YCbCr color space.

Consequently, we decide to choose this method to implement our skin detection scheme. In planar envelope method [7], a pixel is considered as a skin pixel if the color of the pixel satisfies the following two conditions:

Condition 1:

$$S >= Th_{s;} \quad V >= Th_v \ ; \quad S <= -H - 0.1V + 110; \qquad (2.6)$$

$$H <= -0.4V + 75;$$

Condition 2:

$$If \quad H >= 0 \qquad S <= 0.08(100 - V)\, H + 0.5V$$

$$Else \quad S <= 0.5H + 35$$

In [7], Th_s and Th_v are set to 10 and 40, respectively. However, in this study, we found that setting Th_v as 35 can obtain better skin detection results.

Here, (H, S, V) have been translated and scaled to (-180, 180), (0,100), (0,100) respectively based on normalized HSV color space. Figure 2.8 shows one result of detecting the skin tone regions. We observe that not only the skin regions have been detected successfully but also most of non-skin regions in the image have been filtered out.

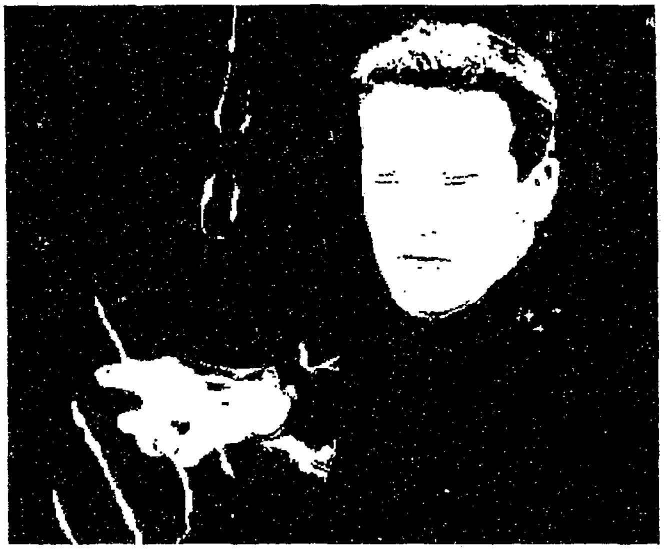


Figure 2.8 Result of skin detection

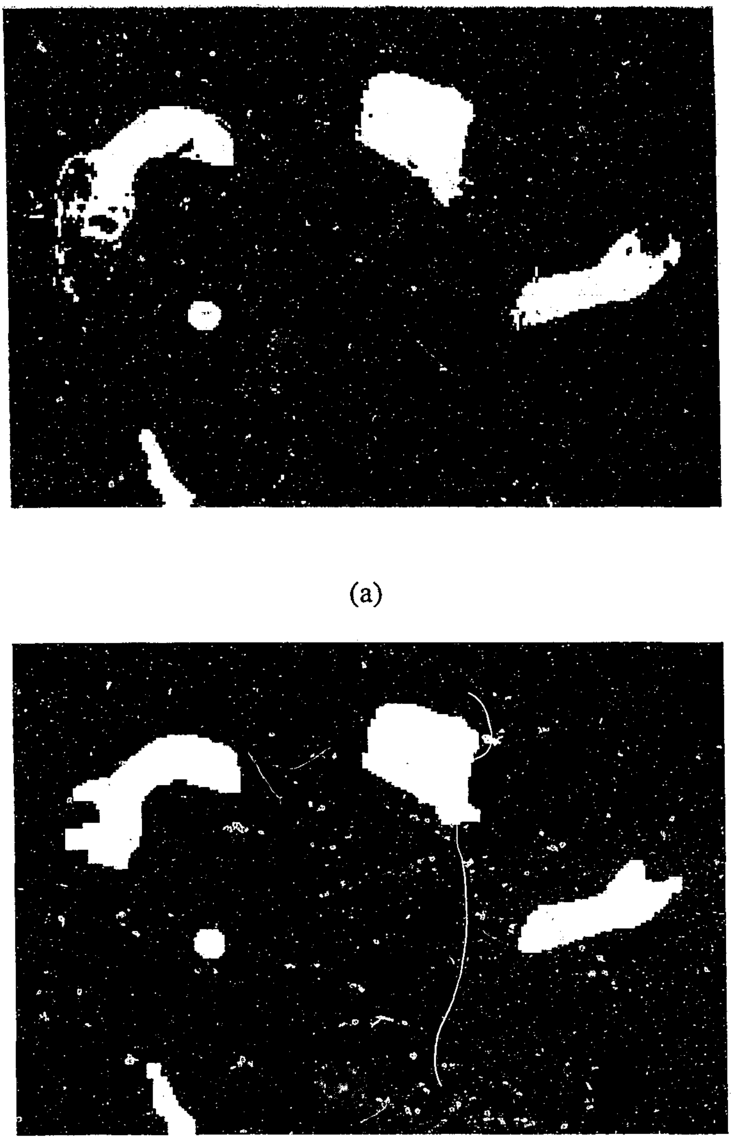
2.2 Cleaning of the skin tone region

After applying the skin segmentation, some non-skin regions such as small isolated blob and narrow belts are inevitably observed in the result as their colors fall into the skin color space. Keeping these spurious skin regions will yield several adverse effects to the latter processing. First, these small isolated regions and narrow strips might be connected to the real face regions. Second, these regions will increase the number of face candidates detected, which will result in expensive computation in the subsequent shape analyses and face verification. Therefore, it is necessary to remove these small isolated areas and narrow belts, and separate some weakly linked regions. We apply the morphological operation to implement above cleaning procedure. The morphological operations consist of opening and closing operations. The opening and closing operations, in turn, depend on the dilation and erosion processes. We describe these four kinds of operations as follows:

- Dilation: The value of the output pixel is the maximum value of all pixels in the input pixel's neighborhood which is defined by the structuring element of dilation operation (such as 5x5 window). In a binary image, if any of the pixels is set to the value 1, the output is set to1.
- Erosion: The value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.
- Opening: erosion of a region followed by dilation. The opening operation breaks the narrows gaps and removes small objects from an image while preserves the shape and size of larger objects in the image.
- Closing: dilation of a region followed by erosion. The closing operation merges regions separated by thin gulf and fills small holes.

The closing operation is first performed to connect narrow gaps between skin regions, and then the opening operation is applied to remove small isolated bulbs and separate the regions connected by thin strips. Finally, we need to perform the 'fill the hole' operation to remove the black isolated holes.

Theoretically, if we want to obtain a better performance, the size of the structuring element of the morphological operation should be proportional to the size of a face. However, the size of a face region in an image is not known until that image is processed through the face detection process. Consequently, we use a fixed size structuring element for the morphological operation instead. In our study, we use the structuring element size of 5x5. The choice of this size seems to give the best performance.



(a)

(b)

Figure 2.9 (a) Result of skin detection

(b) Result of cleaning of skin tone regions

Figure 2.9 shows the result of performing morphological operations. We can observe that not only have the isolated tiny bulb been eliminated and the number of face candidate regions been reduced but also the boundaries of the regions are smoothed out.

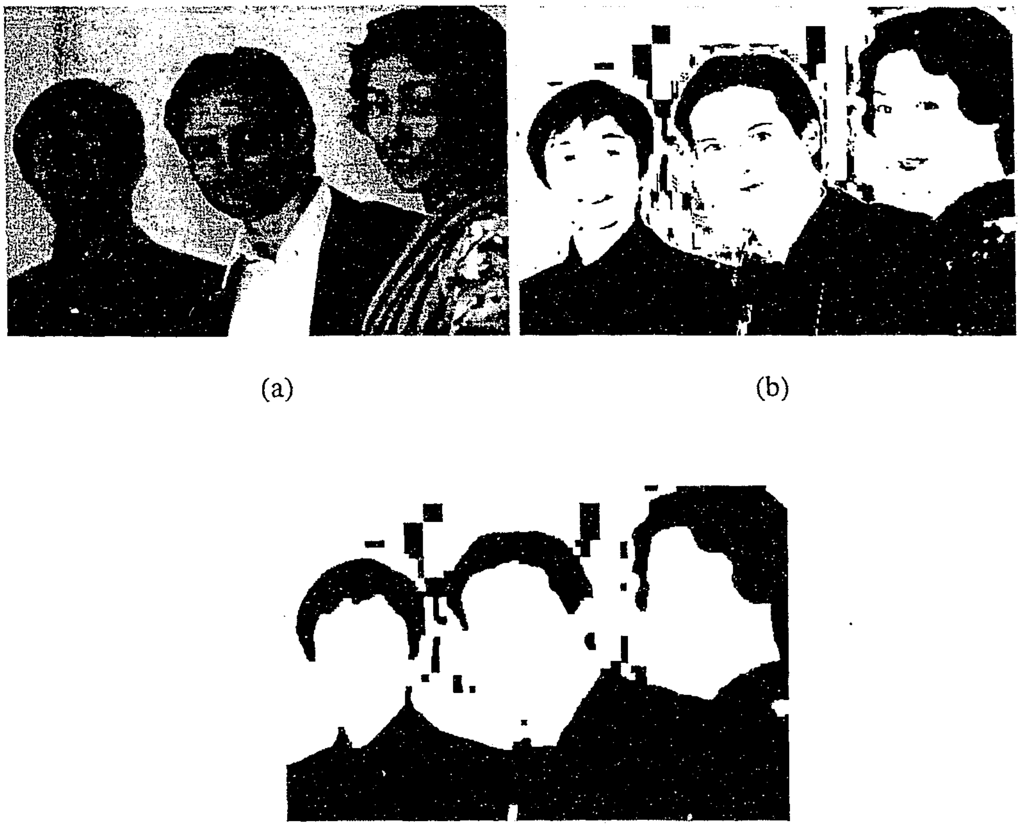
2.3 Unsupervised color segmentation

After performing skin color detection and cleaning of the skin color region, we can extract some face candidate regions from background successfully. However, when some other objects or the background contain the color similar to skin tone, the actual face region will be connected to those object regions or background, as it is shown in Figure 2.10. Therefore, we employ a color segmentation scheme to cope with such situation. As this color segmentation scheme will automatically provide the number of clusters in the color space, we call it unsupervised color segmentation. The color segmentation is the process of classifying the pixels within a multi-channel image into a set of clusters with a uniform color characteristic and dividing the color image into meaningful objects. Among the various approaches to color segmentation, multi-channel clustering has shown to be an effective method for color image segmentation in various color space [8,17,19]. Here, we use the HSV color space to partition an image into different regions.

In HSV color space, the Hue component is the most significant feature that can be used to detect uniform color regions [8]. However, the Hue can be unreliable and unstable under following situations:

1. The level of brightness (i.e. Value) in the image is low, which corresponds to the shadows and low lighting level parts of the image.
2. Some areas have rather low Saturation values, which are found in the achromatic regions of the image. We usually use black, gray, and white to describe the achromatic regions.

In order to perform the appropriate segmentation scheme that could deal with above situations, the image should be separated into the achromatic areas and chromatic areas. Several kinds of thresholds for Hue, Saturation, and Value have been selected to partition the achromatic regions and chromatic regions [8,17].



(a) (b)

(b)

Figure 2.10 (a) Original image (b) Result of skin detection

(c) Result of cleaning skin

According to [8], we define the effective ranges of the hue, saturation and value in the HSV space model and divide the color space into three regions: chromatic region(S > 20% and V> 25%), achromatic region(S <=10% or V<= 25%) and transitional regions (10% < S <20% and V > 25%). As the Hue component has great discriminating capability for the color component, it is used to segment the chromatic regions of the image, while the Value component is employed to segment the achromatic regions of the image. In the transition region, the Hue component again becomes unreliable; therefore, the pixels of this region are better specified based on the Value component information. Also in this area, the Saturation component could be considered for a further refined segmentation.

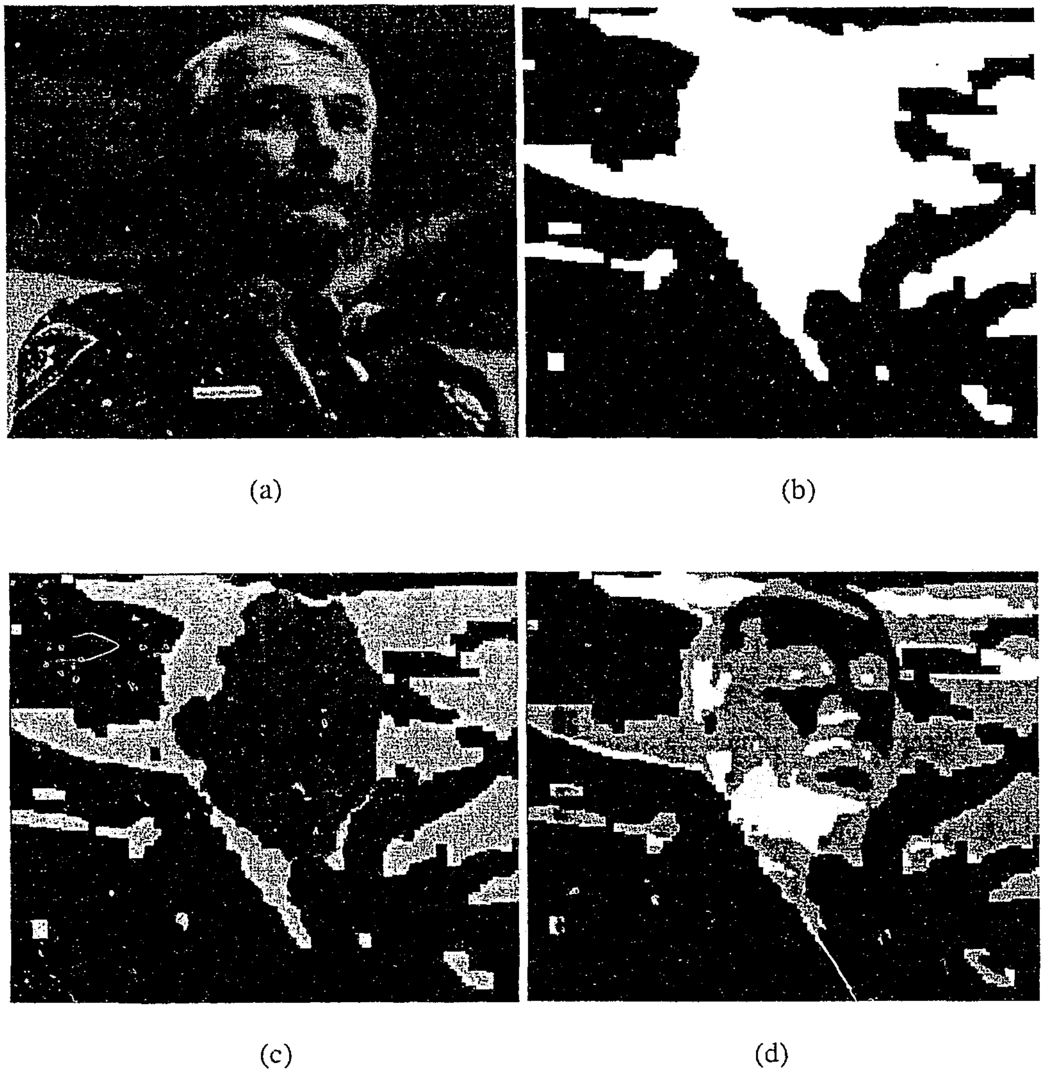
Above we discuss the fixed thresholds for Saturation and Value to separate the chromatic area and achromatic area. An adaptive threshold for Saturation has presented in [23] for S in the range [0.1]:

$$Th_{sat}(V) = 1.0 - 0.8*V \qquad (2.7)$$

Here, the range of V is within [0, 1] and $0.2 \leq Th_{sat} \leq 1.0$.

The Saturation threshold determines whether a pixel should be specified by the chromatic information or intensity information. If the Saturation of a pixel is larger than the threshold, the pixel should be presented by its Hue and belongs to the chromatic area, otherwise, the pixel is presented by its intensity and belongs to the achromatic area. The threshold of the Saturation component is dependent on the Value component. For the situation of low intensities, even if the Saturation is high, a color is still considered as the gray value and specified by Value component.

Next, we compare the results of color segmentation of two threshold methods. The original image is shown in Figure 2.11 (a). Figure 2.11 (b) shows the result of skin tone detection. Figure 2.11 (c) shows segmentation result for the fixed threshold method. Figure 2.11 (d) shows segmentation result for the adaptive threshold method. We can observe that the adaptive threshold method can not separate skin region from the background which has the color similar like skin. Furthermore, the adaptive threshold method might introduce another problem: it divides the face candidate region into many smaller regions. As the result, we must add another extra stage of region merging to solve this problem. Therefore, we decide to choose the fixed threshold method to perform the color segmentation.



(a)

(b)

(c)

(d)

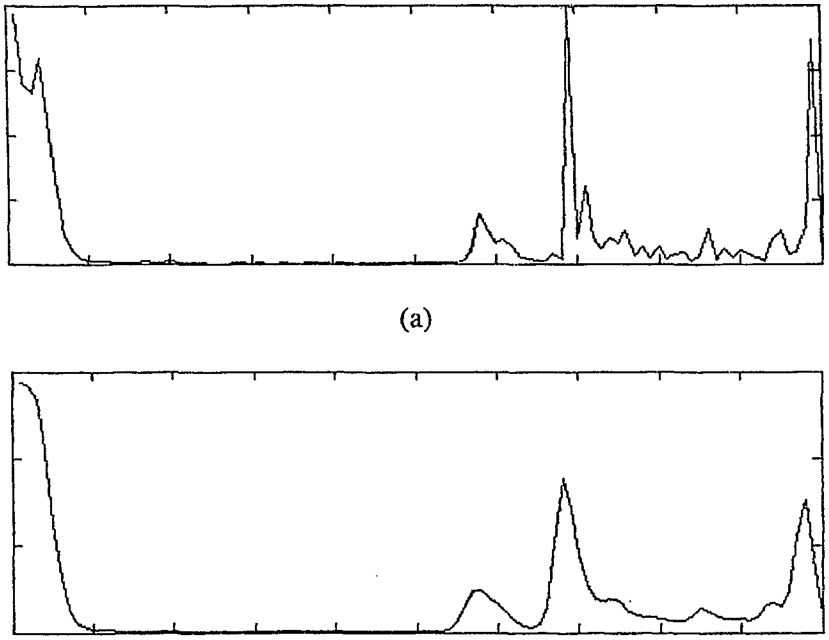
Figure 2.11 (a) the original image (b) the result of skin detection (c) Segmentation result of the fixed threshold method (d) Segmentation result of the adaptive threshold method

2.3.1 Histogram clustering

After the image has been divided into the chromatic areas and achromatic areas, a clustering procedure is implemented in corresponding area. In clustering technique, a histogram is first obtained by the color value at all pixels and the shape of each cluster is found. Then, each pixel in the image is assigned to the cluster that is closest to pixel color in space. For our situation, we first perform the segmentation in the chromatic regions on the clustering of the hue histogram. Prior to the clustering process over the Hue histograms, it is necessary to smooth them out to remove the histogram estimation error. To that end, the scale-space filter is employed. The similar operation was also found in [8,19]. The smoothed histogram is obtained by convolving the original histogram h(x), with a Gaussian function g(x, τ), this Gaussian function has zero mean and standard deviation τ, which is also called the scale constant and can be applied to control the smoothing degree of the histogram. The convolution is given by

$$\begin{aligned} F_h(x,\tau) &= h(x) * g(x,\tau) \\ &= \int_{\infty}^{\infty} h(u) \frac{1}{(2\pi)^{1/2}\tau} \exp\left[-\frac{(x-u)^2}{2\tau^2}\right] du \end{aligned} \tag{2.8}$$

Where "* " implies a 1-D convolution. $F_h(x, \tau)$ represents the smoothed histogram. Figure 2.12 shows the Hue histogram of Figure 2.7 (a) and its smoothed version.



(a)

(b)

Figure 2.12 Hue histogram of Figure 2.7 (a) and its smoothed version.

After obtaining the smooth histogram, we use a watershed based techniques [21,27] to cluster the hue histogram. First, H_c, the complement of Hue histogram H_h is computed. From H_c, all the minima are extracted and used as the seeds for the 1-D watershed. This results in splitting of the histogram into several regions. Then, pixels which belong to these regions will be assigned to the different labels respectively. The watershed method had two advantages:

(1). the method provides the number of the clusters automatically;

(2). the algorithm is very efficient in computation.

The drawback of this algorithm is that it usually yields an excessive number of the cluster: an over-segmentation of the histogram. We adopt two steps tu reduce the over-segmentation.

First, we simplify the original histogram by reconstructing a new histogram. The new histogram is obtained as follows:

$H_n(n) = H_o(n) - \delta(H_o)$

If $H_n(n) <= 0$ then $H_n(n) = 0$.

Where $H_n(n)$ and $H_o(n)$ represent the new histogram and old histogram, respectively. $\delta(H_o)$ denotes the standard deviation of the old histogram.

We then use more restrictive criteria, normalized contrast [21], to remove the insignificant local minimum points. The definition of normalized contrast is expressed by the following formula:

normalized contrast = Contrast/Height

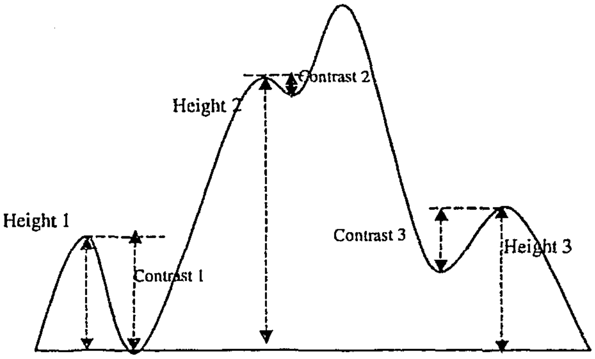
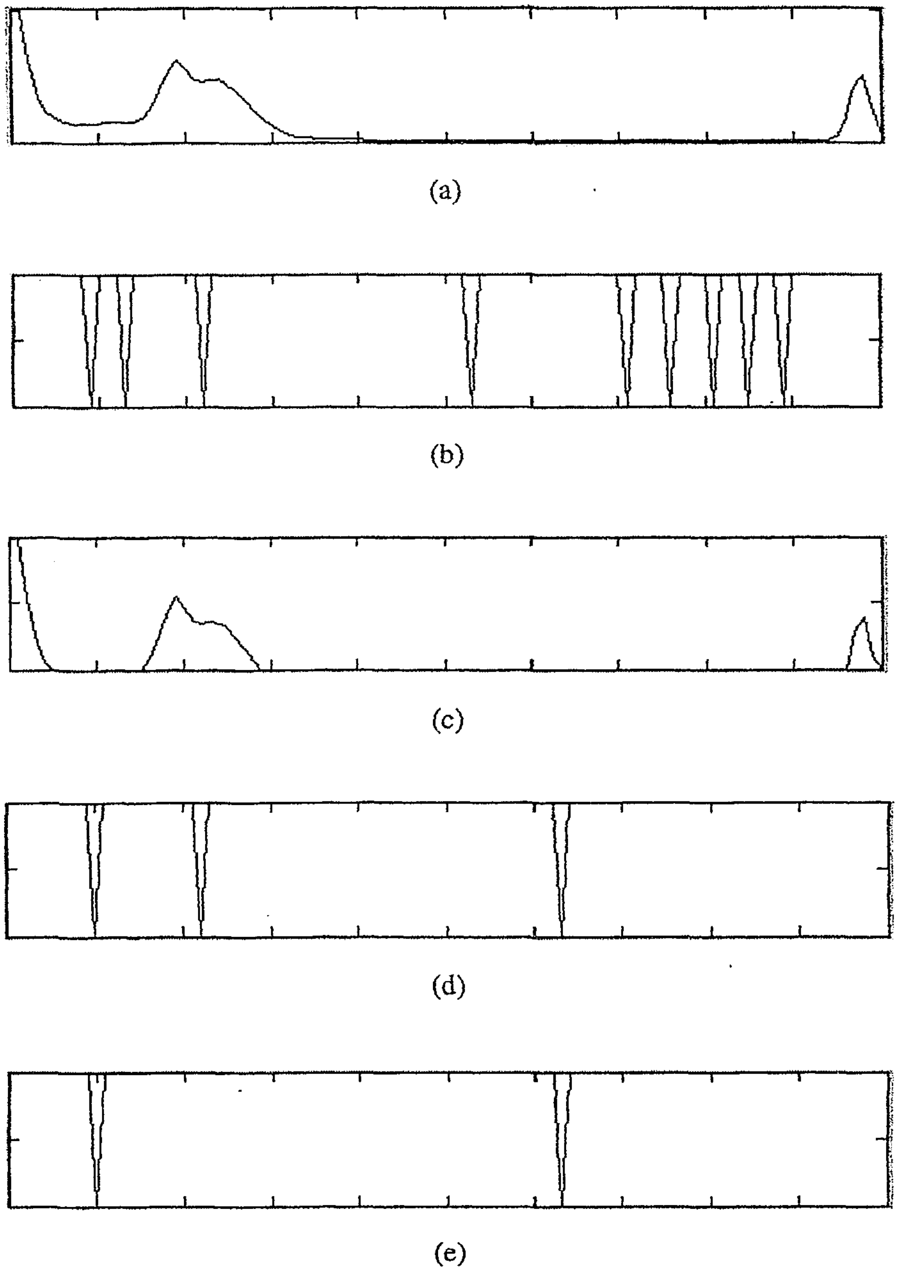


Figure 2.13 Height and contrast for a histogram peak

Where *Height* is the height of the local maximum and *Contrast* is defined as the height difference between a maximum and the higher of its two local minima. Figure 2.13 illustrates *Contrasts* and *Heights* of several local maxima. If the normalized *Contrast* is less than a suitable threshold such as 10%, the higher minimum of the two minima will be removed. The result is that the regions partitioned by that minimum will be merged.



(a)

(b)

(c)

(d)

(e)

Figure 2.14 An illustration of reducing the over-segmentation

We use Figure 2.14 to illustrate the reduction of over-segmentation. Figure 2.14 (a) is a smoothed histogram. Figure 2.14 (b) is the result of watershed from Figure 2.14 (a). Zero points represent the minimum points of the histogram. Figure 2.14 (c) denotes the simplified histogram after reconstruction. Figure 2.14 (d) is the result of watershed from Figure 2.14 (c). Figure 2.14 (e) represents the final minima plot after considering the normalized contrast criteria. We observe that the number of the regions on the histogram has been reduced effectively.

The same operations of smoothing histogram and the watershed base approach will be performed for the value histogram in the achromatic regions and transitional regions.

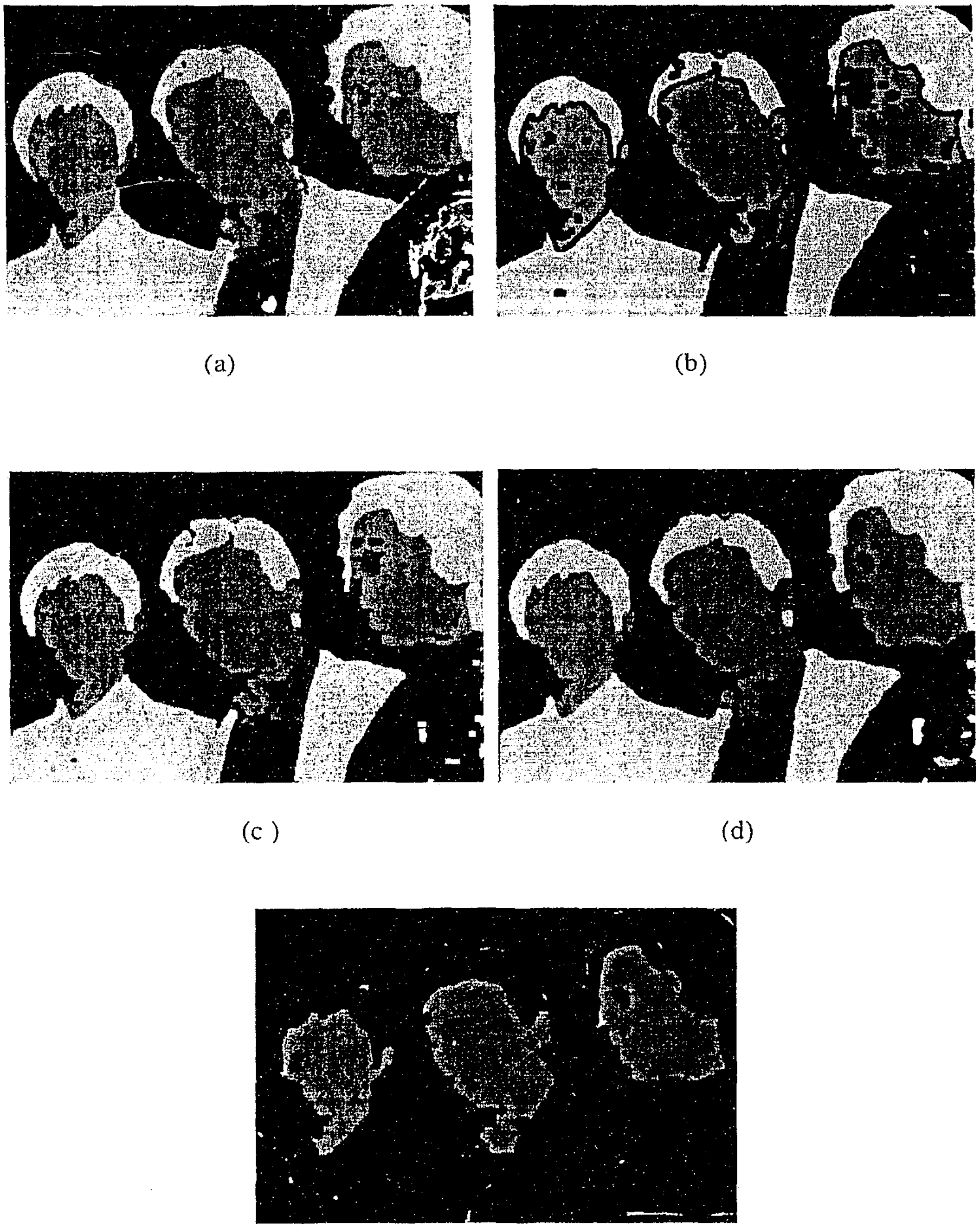
2.3.2 Label morphological filtering

After performing the color segmentation, median filtering and label morphological erosions and dilations will be applied in order to refine the results obtained from the initial segmentation. The label morphological operators [21] are required to remove spurious regions and isolated small regions, and make the border of two regions clearer. We describe the procedure of label morphological filtering as following:

(1) Label erosion: if a pixel has a label j and any of its neighbors has a label k ≠ j, then pixel is set to unassigned.

(2) Label dilation: if all the assigned pixels in the neighborhood of an unassigned pixel have the same label j then the label j is assigned to the unassigned pixel.

According to actual situation, label erosion and label dilation will be implemented repeatedly in order to obtain the better processing results.



(a)

(b)

(c)

(d)

(e)

Figure 2.15(a) Segmentation after histogram clustering. (b) Label erosion on (a) twice. (c) Label dilation on (b) twice. (d) The result of Flood fill algorithm. (e) The final segmentation

Figure 2.15 (a) is the resulting image of the color segmentation from Figure 2.10 (a). Figure 2.15 (b) shows the result of applying the label erosion on Figure 2.15 (a) twice. Figure 2.15 (c) shows the result of applying the label dilation on Figure 2.14 (b) twice.

Once the label morphological filtering is finished some appropriate labels will be used to fill the unassigned pixels (black areas). We apply two methods to implement this operation.

a) Maximum histogram method: For each unassigned pixel, we first obtain the local window (such as 3x3 window) histogram of the label image. Then we use the label which possesses the maximum value in local histogram to fill the unassigned pixels.

b) Label flood fill method: This process is operated by propagating the labels until no unassigned pixel remains according to spatial coherence relationship between the unassigned pixels and the nearest assigned pixels. The algorithm can be illustrated as following:

   1) Find out all unassigned pixels.
   2) For each unassigned pixel, check whether there is any assigned pixels in the neighborhood of this unassigned pixel. If there is, mark the unassigned pixel by the label of one of the neighborhood pixels.
   3) Set all marked pixels as corresponding label.
   4) Go to step 1 until no unassigned pixel remain.

By comparing the two methods, we found that Flood fill algorithm can yield more satisfied results. We can combine above two approaches by using Maximum histogram method to obtain a label to mark the unassigned pixel in the second step of

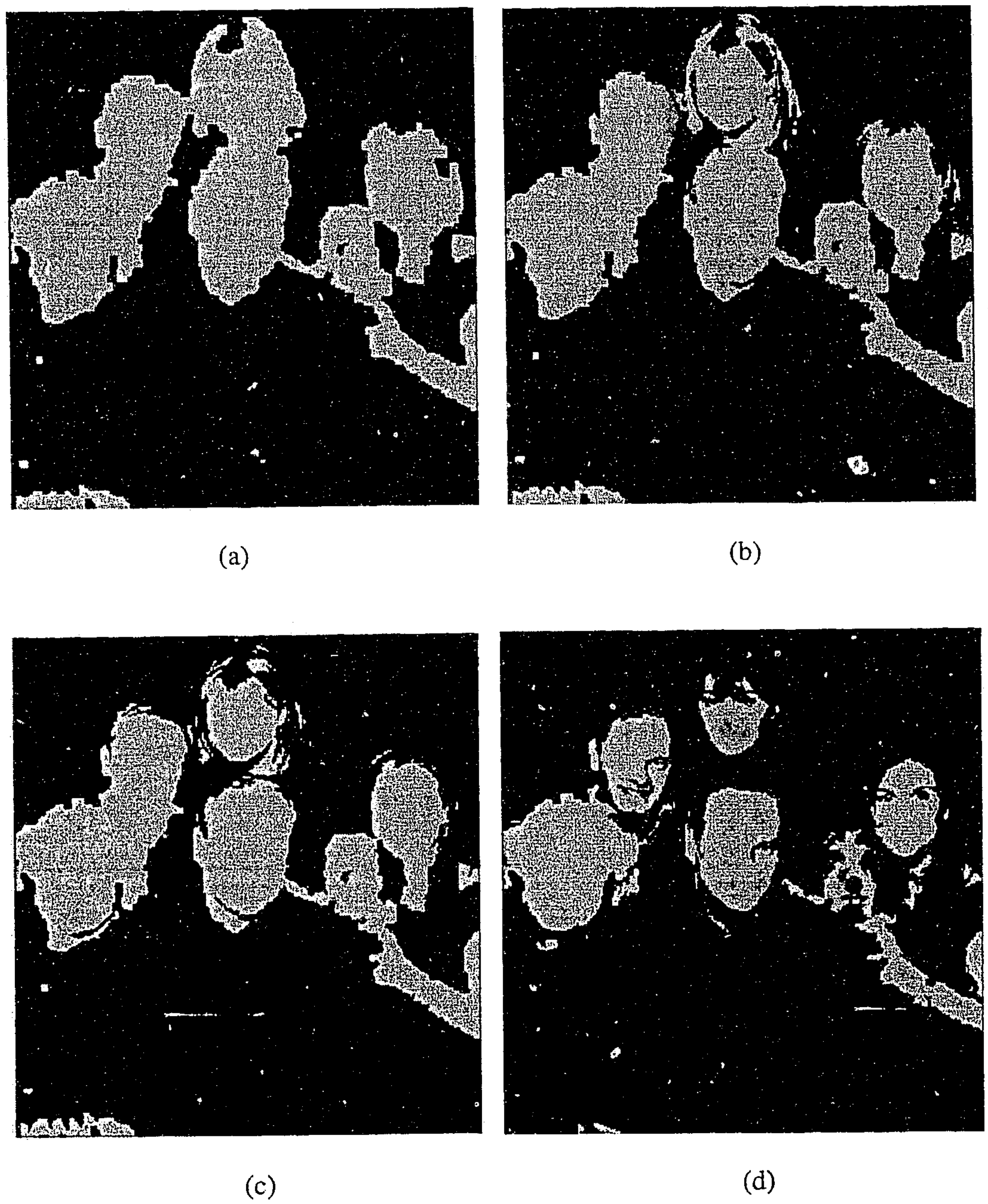
Flood fill method. Figure 2.15 (d) shows the result of Flood fill algorithm in which all unassigned pixels have been removed.

Finally, the results of color segmentation and skin detection are combined by an AND (multiplication) operation. Figure 2.15 (e) shows the resulting segmentation image. After this stage, we can begin to perform the operation of extracting the potential face candidate regions.

2.4 Iterative luminance segmentation

After applying the skin detection, morphological operation, and unsupervised color segmentation, the detected skin region could still be linked together or attached to other skin-colored object such as hair, clothes, leather, and wood. Above situations are very likely to happen especially in complex background environments or multiple faces situation. Figure 2.16 (a) shows an example of this situation.

For human eye, it is not difficult to discriminate two skin colors and connected objects based on the boundaries between them, the reason is that in the small neighborhood of the boundaries of objects the luminance component always have high variance. By simulating this human intuitive detection, we perform the binary segmentation based on the luminance component of the image. However, the selection of the threshold for luminance segmentation is a difficult task since different images, or even different parts in the same image requires different threshold. Here, we adapt iterative segmentation by increasing threshold gradually. Figure 2.16 shows the several different stages in iterative luminance segmentation. We can see that after stages of skin detection and color segmentation, several face regions are still linked together. During the process of iterative luminance segmentation, each face region has been gradually separated or detached from other face regions, skin colored objects, or background.



(a)

(b)

(c)

(d)

Figure 2.16 Several different stages in iterative luminance segmentation for increasing threshold gradually

2.5 Location of candidate face region

The purpose of this stage is to remove as many as possible the non potential face candidates that might have been falsely detected and remained after the processing in previous stages. We will first use constrains corresponding to size, shape, symmetry, and color homogeneity to remove most of the non potential face candidate regions.

(1) Size

In most cases, we are only interested in a relatively large face. Therefore, the very small size regions can be removed. We set the lower-bound of the region size as 1/150 of the image. For some specific applications such as survey, we may want to set the lower-bound of the region size as the absolute value such as 15x15. It is natural that the upper-bound of the region size is set as the whole frame image.

(2) Shape

The shape of the human face is a much discriminated criteria. We distinguish the face shape by checking the aspect ratio of its bounding box. The bounding box of a region is the minimum rectangle which contains all pixels of the region. We set the range of the aspect ratio as [0.7, 1.9] based on experimental results. Considering the different orientation and pose of the faces, the range of the aspect we select is deliberately large in order to assure that the potential face candidates are not rejected.

(3) Symmetry

Generally, the shape of human face is rather symmetrical. We divide the bounding box of face candidate region into four equal smaller parts with a vertical line and a horizontal line that cross the center of the region. Then we check for each sub-region to determine whether the ratio of pixels within the sub-region to ¼ bounding box is larger than a threshold such as 0.25. If it is not, that face candidate region will be discarded.

(4) Density of the skin

The Density of skin is the number of skin color pixels divided by the total number of pixels in the bounding box. We use the density of the skin to judge the color homogeneity of the face candidate regions. According to [7], we divide the bounding box into two parts, the outer and inner areas. The outer area is referred as the border area of the bounding box, whose width is 15% of the bounding box width, then, the rest of the bounding box is referred as the inner area. If a face candidate region is considered as a potential face region, the density of the skin in the outer area and inner area should be large than corresponding thresholds such as Th_o and Th_i. Since the outer area might include the hair and part of background and the inner area mainly consists of the skin part of the face, the Th_o should be larger than Th_i. As in [7], we chose two thresholds as following:

$$Th_o = 0.5,$$

$$Th_i = 0.7 * Th_o + 0.3$$

After performing above operations, a significant number of regions are rejected from the set of face candidate regions. But there are still quite a few false positive regions existing. We observe that because of existence of face features such as eyes, mouth, and nose as well as other parts of the face, the real face region always demonstrates higher variance in the luminance components of the image comparing with other skin color region. Therefore, we could make use of this useful characteristic to discriminate the face region from other skin color objects. The criterion for discrimination is the ratio of the pixels high variance neighborhood to the total number of pixels of the skin color region. The skin region must have been processed by the morphological closing operation in order to fill the hole inside the skin color region. When the ratio is larger than a threshold, we accepted the face candidate region as a face region, otherwise, that region will be removed.

Considering that most of faces we want to detect is upright and front, we apply the Sobel horizontal edge detector to obtain pixels with high variance neighborhood. Figure 2.17 gives an example of this feature. Figure 2.17(a) shows the skin color region detected. Figure 2.17(b) shows the parts of the image which have high variance in intensity.

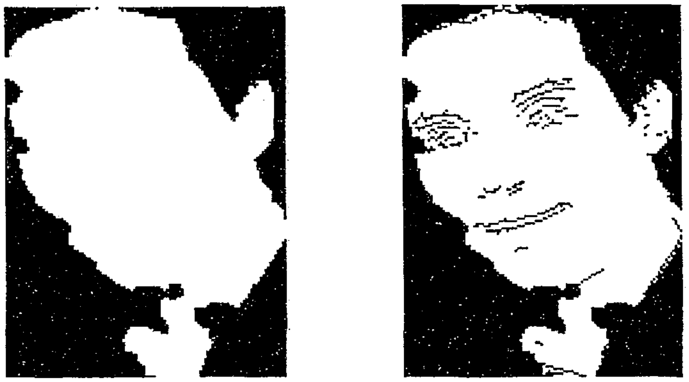


Figure 2.17 (a) the skin region detected

(b) The pixels with high difference in intensity.

Chapter 3

Experimental Results

The face detection scheme proposed has been trained with 10 images which contain 25 faces and tested with 10 images which contain 21 faces. These two sets of images were randomly extracted from the internet and personal photo collection. These images contain multiple faces with variations in lighting conditions, colors, positions, scales and orientations. Some of them have been taken with complex backgrounds.

The algorithm is implemented by matlab. For a 250x220 image, the processing time is less than one minute. For a 401x410 image, the processing time is less than two minutes. Most of the processing time is consumed in the stage of color segmentation and iterative luminance segmentation. The processing time can be decreased through optimizing code.

The experimental results are shown in table 3.1. For the 10 training images, all 25 faces are detected correctly; for the 10 test images, 18 out of 21 faces are detected correctly, but at the same time, there are some false positive faces detected.

False positive results from some detected regions having color similar to skin tone, shape and texture of the face. False negative is mainly caused by the partially occluded face, extreme lighting condition, or detected face connected with other parts of body such as arms and the chest.

Table 3.1

The results of face detection

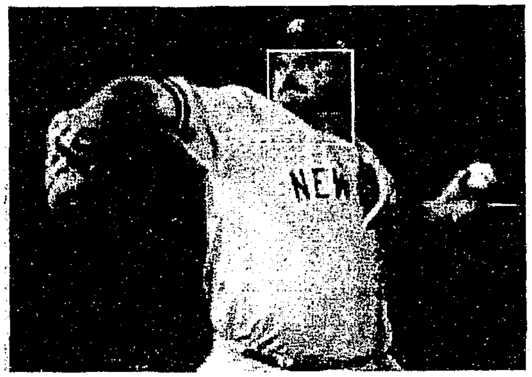
(a) Training Results

Number of images	Number of faces	Correctly detected	False positive	False negative
10	25	25	3	0

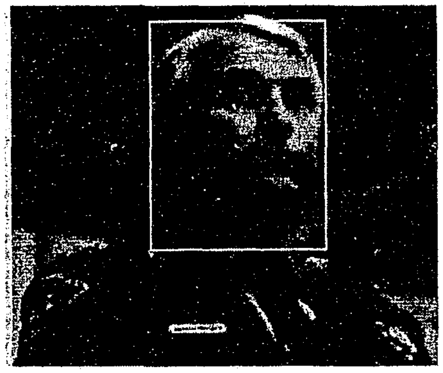
(b) Testing Results

Number of images	Number of faces	Correctly detected	False positive	False negative
10	21	18	8	3

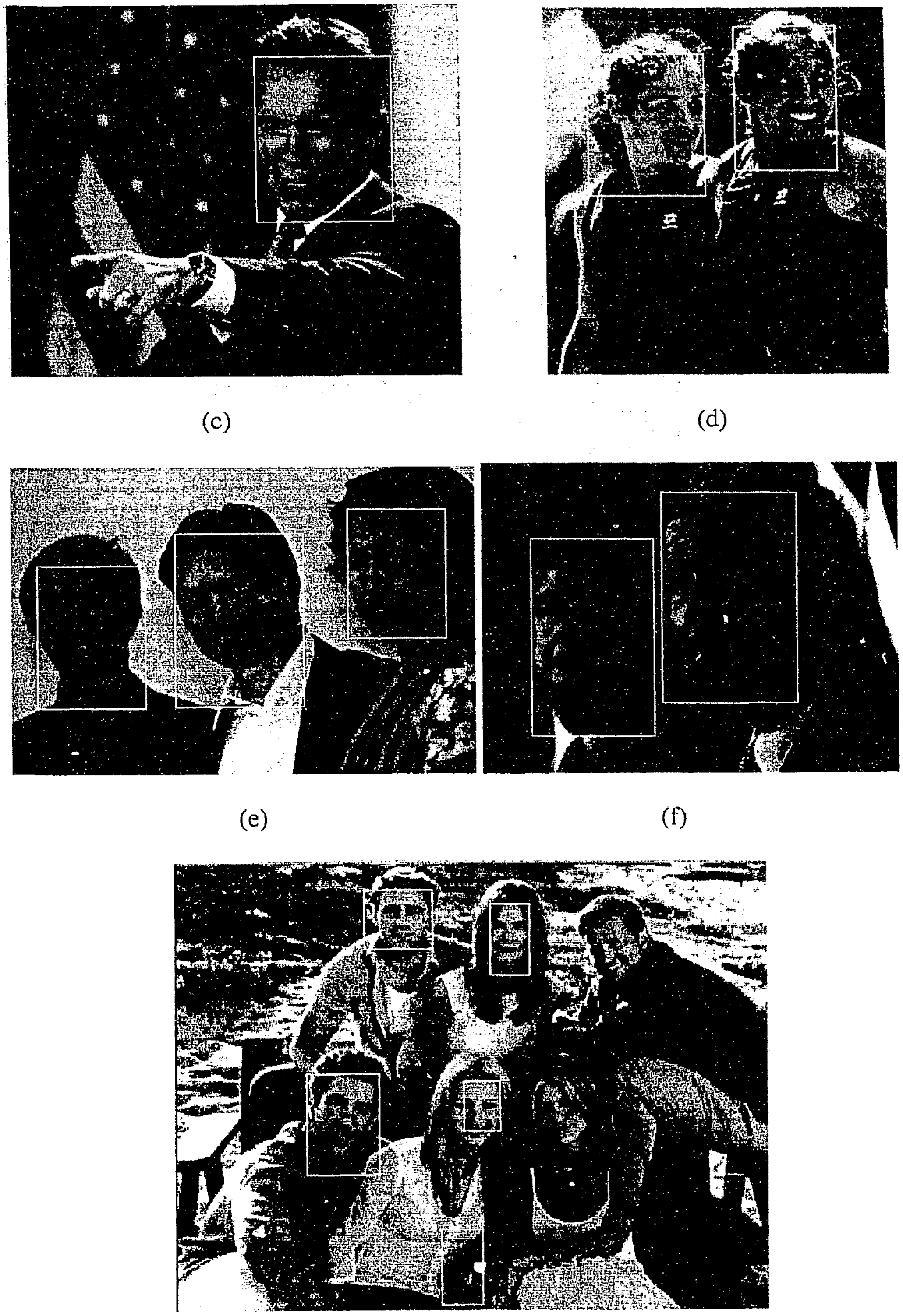
Some results of proposed face detection scheme are shown in Figure 3.1. Color images with multiple faces in various size, different colors and position are presented in these examples. Some false positive and false negative example can also be observed in detected results. We also observe that even the proposed algorithm is designed to primarily detect faces that are upright and frontal; it is also suitable for slightly tilted orientation of faces.



(a)



(b)



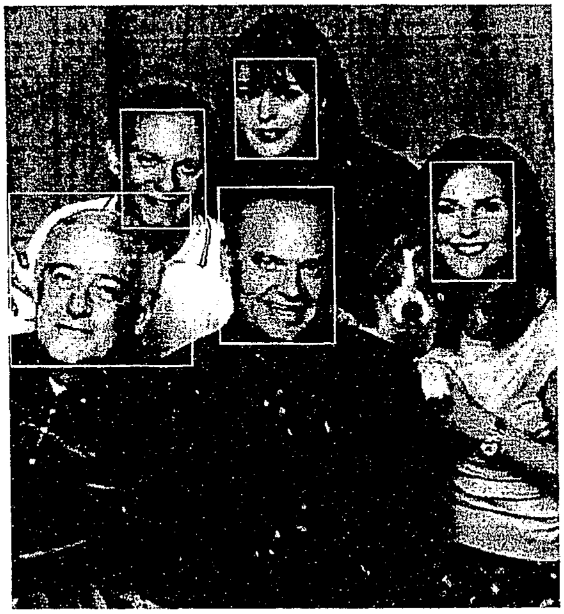
(c)

(d)

(e)

(f)

(g)



(h)

Figure 3.1 Results of proposed face detection scheme

Chapter 4

Conclusion and Future Work

In this study, a face detection scheme has been presented. The scheme is designed to operate on color images. In the first stage of algorithm, the skin color regions are detected based on the chrominance information. An unsupervised color segmentation stage is employed to make skin color regions to be divided into smaller regions which have homogenous color. Then, we employ the iterative luminance segmentation to further separate the objects with similar skin color based on the high variance of the luminance component in the neighborhood of edges of objects. Post-processing is applied to determine whether skin color regions fit constrains of face on density of skin, size, shape, and symmetry and these regions contain the facial features such as eyes and mouth. Experimental results show that the algorithm is robust and is capable to detect the even multiple faces in the presence of a complex background which contains the color similar to the skin tone.

The proposed face detection scheme is far from perfect. Further works are required to improve the performance. One possible approach is to develop a more accurate method to detect facial features, such as identification of eyes and mouth [3].

Bibliography

[1].Ming-Hsuan Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: a survey", IEEE *Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp 34 – 58, Jan. 2002.

[2]. M. Abdel-Mottaleb and A. Elgammal, "Face detection in complex environments from color images", *IEEE Int'l Conf. Image Processing*, ICIP 99, vol. 3, pp 622 – 626, Oct. 1999.

[3]. R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images,'" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696-706, May 2002.

[4]. Ming-Hsuan Yang and N. Ahuja, "Detecting human faces in color images", *IEEE Int'l Conf. ICIP 98, Image Processing*, vol. 1, pp127-130. Oct. 1998.

[5]. J. Karlekar, J and U.B. Desai, "Finding faces in color images using wavelet transform" " *IEEE Int'l Conf. Image Analysis and Processing*, pp1085-1088, Sept. 1999

[6]. Huynh-Thu Quan, M. Meguro, and M. Kaneko, "Skin-color extraction in images with complex background and varying illumination", *Proceedings. Sixth IEEE Workshop, Applications of Computer Vision, 2002*, pp 280-285, Dec. 2002.

[7]. C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis", *IEEE Trans. Multimedia*, vol. 1, pp 264-277, Sept. 1999.

[8]. N. Herodotou, K.N. Plataniotis, and A.N. Venetsanopoulos, "A color segmentation scheme for object-based video coding", *IEEE Symposium, Advances in Digital Filtering and Signal Processing*, pp 25-29, June 1998.

[9]. Gang Wei and K. S. Ishwar, "Face detection for image annotation", *Pattern Recognition letters 20*, pp1313-1321, 1999.

[10]. Quan Yuan, Wen Gao, and Hongxun Yao, "frontal face detection in complex environment", *IEEE Int'l Conf. Pattern Recognition*, vol. 1, pp25-28, Aug. 2002.

[11]. Haiyuan Wu, Qian Chen, and M. Yachida, "Detecting human face in color images" *IEEE Int'l Conf. Systems, Man, and Cybernetics*, vol.3, pp2232-2237, Oct. 1996

[12]. J. Cai, A. Goshtasby, and C. Yu, "Detecting human faces in color images", *IEEE Proceedings, International Workshop, Multi-Media Database Management Systems*, pp124 –131, Aug. 1998.

[13]. H. Wu, Q. Chin, and M. Yachia, "Face Detection from Color Images using a Fuzzy Pattern Matching Method", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp 557-563, June 1999.

[14]. M. Kirby and L. Sirovich, "Application of Karhunen-Loeve Procedure for the Characterization of Human Faces", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp 103-108, Jan. 1990.

[15]. H. A. Rowley, S. Baluja, and T. Kanade, "Neural Network based Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp 23-38, 1998.

[16]. A. Lanitis, C.J. Tayor, and T.F. Cootes, "An Automatic Face Identification System using Flexible Appearance Models", *Image and Vision Computing*, vol. 13, pp 393-401, 1995.

[17]. Din- Chang Tseng and Chung-Hsun Chang, "Color Segmentation Using Perceptual Attributes" *Proceedings, 11th IAPR Int'l Conf. Image, Speech and Signal Analysis,* pp. 228-231, 1992.

[18]. G Gregory A. Hance, Scott E. Unbough, Randy H. Moss and William V. Stoecke, "Unsupervised Color Image Segmentation", *IEEE Engineering in Medicine and Biology* pp. 104-111, 1996.

[19]. L in YW and Lee SU, "On the Color Image Segmentation Algorithm Based on The Thresholding and The Fuzzy c-Means Techniques", *Pattern Recognition,* 23(9): 935-952, 1990.

[20]. Serge Beucher and Freinand meyer, Mathematical Morphology in Image Processing, chapter 12. The morphological approach to Segmentation: The Watershed Transformation pp. 433-481, Marcel Dekker Inc., 1993.

[21]. A. Alberto Albiol, Luis Torres and Edward J. Delp, "an Unsupervised color Image Segmentation Algorithm for Face Detection Applications", *IEEE Int'l Conf. Image Processing, Proceedings,* vol. 2 pp 681-684, Oct. 2001.

[22]. M. Petrou, L. Shafarenko and J Kittle, "Histogram-based segmentation in a perceptually uniform color space", *IEEE Trans Image Processing*, vol. 7, pp1354-1358, 9 1998.

[23]. S. Sural, G. Qian, and S. Pramanik, "Segmentation and Histogram Generation using the HSV Color Space for Image Retrieval", *IEEE ICIP 2002*, pp589-592