FACE DETECTION

Sahar Noor Abdal
ID: 05310049
Mashook Mujib Chowdhury
ID: 05310052

Department of Computer Science and Engineering
January 2008

DECLARATION

We hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Supervisor:

Dr. Tarik Ahmed Chowdhury

Assistant Professor

Computer Science & Engineering

Students:

Sahar Noor Abdal

ID: 05310049

Mashook Mujib Chowdhury

ID: 05310052

ACKNOWLEDGMENT

We would like to thank our supervisor,

Professor Dr. Tarik Ahmed Chowdhury for his patience, guidance and expertise throughout this thesis.

We are also grateful to Henry Chang and Ulises Robles for their project on “Face Detection’” as we have implemented their algorithm and used their project as our key reference.

ABSTRACT

Human face detection plays an important role in applications like face recognition, video surveillance, human computer interface, face image database management and many more. In modern multimedia systems, video and image signals usually need to be indexed or retrieved according to their contents. In our thesis, we implement a color characteristic for use in detection of frontal human faces in color images with complex backgrounds i.e. a color based technique to detect frontal human face had been developed and implemented. A technique for detecting frontal human faces in color images is described that first separates skin region from non-skin region and then locates faces within skin regions. Using color information in an image is one of the various possible techniques for face detection. The technique involves conversion of a color image into a gray scale image in such a way that the gray values in the pixel shows the likelihood of the pixel belonging to the skin. Obtained gray scale image is then segmented to skin and non-skin regions, and a model face, representing front face is used in template matching process to detect face within skin regions i.e. to find which of the candidates is/are actually a face. Later, the false-positive and false-negative errors of the implemented face detection technique on color images are calculated. The experimental results show that this method can detect faces in

the images from different sources with good efficiency. Since faces are common elements in video and image signals, the proposed face detection technique is an advance towards the goal of content-based video and image indexing and retrieval.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

FACE DETECTION

Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a ‘specific’ case of object-class detection, where the task is finding the location and sizes of all objects in am image that belong to a given class.

Face detection, can be regarded as a more ‘general’ case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one) [1].

There are different approaches to face detection. This means that there are a number of approaches to detect a face in a scene. They include both the easy approach and the difficult approach. Below there is a list of most *common approaches* to face detection [2].

1.1 Finding faces in image with controlled background

This is the easiest way out and easy of all the approaches. In this approach, images are used with a plain mono color background, or images with a predefined static background. As removing the background gives the face boundaries.

1.2 Finding faces by color

This is the approach where face is detected using skin color. Once we have access to color images it is possible to use the typical skin color to find face segments. But in this approach, there is a drawback. Skin color varies from race to race and this does not work well with all kind is skin color. In addition, this approach is not very robust under varying lighting conditions.

1.3 Finding faces by motion

By using a real time video it is possible to find face, the face that is always in motion in reality. And in this process a real time video is used to find face. But there is a drawback to this process. Problem arises when other objects are moving in the background.

1.4 Finding faces in unconstrained scenes

This approach is the most complicated approach of all and this approach tops all the other approaches. In this approach, face has to be detected from a black and white still image.

CHAPTER II

EVOLUTION OF FACE DETECTION

Early efforts in face detection have dated back as early as the beginning of the 1970s, where simple heuristic and anthropometric techniques [3] were used. These techniques are largely rigid due to various assumptions such as plain background, frontal face—a typical passport photograph scenario. To these systems, any change of image conditions would mean fine-tuning, if not a complete redesign. Despite these problems the growth of research interest remained stagnant until the 1990s [4], when practical face recognition and video coding systems started to become a reality. Over the past decade there has been a great deal of research interest spanning several important aspects of face detection. More robust segmentation schemes have been presented, particularly those using motion, color, and generalized information. The use of statistics and neural networks has also enabled faces to be detected from cluttered scenes at different distances from the camera.

Additionally, there are numerous advances in the design of feature extractors such as the deformable templates and the active contours which can locate and track facial features accurately.

Because face detection techniques requires *a priori* information of the face, they can be effectively organized into two broad categories distinguished by their different approach to utilizing face knowledge. The techniques in the first category make explicit use of face knowledge and follow the classical detection methodology in which low level features are derived prior to knowledge-based analysis [5,6]. The apparent properties of the face such as skin color and face geometry are exploited at different system levels.

Typically, in these techniques face detection tasks are accomplished by manipulating distance, angles, and area measurements of the visual features derived from the scene. Since features are the main ingredients, these techniques are termed the ***feature-based approach***. These approaches have embodied the majority of interest in face detection research starting as early as the 1970s and therefore account for most of the literature reviewed in this paper. Taking advantage of the current advances in pattern recognition theory, the techniques in the second group address face detection as a general recognition problem.

Image-based [7] representations of faces, for example in 2D intensity arrays, are directly classified into a face group using training algorithms without feature derivation and analysis. Unlike the feature-based approach, these relatively new techniques incorporate face knowledge implicitly [8] into the system through mapping and training schemes.

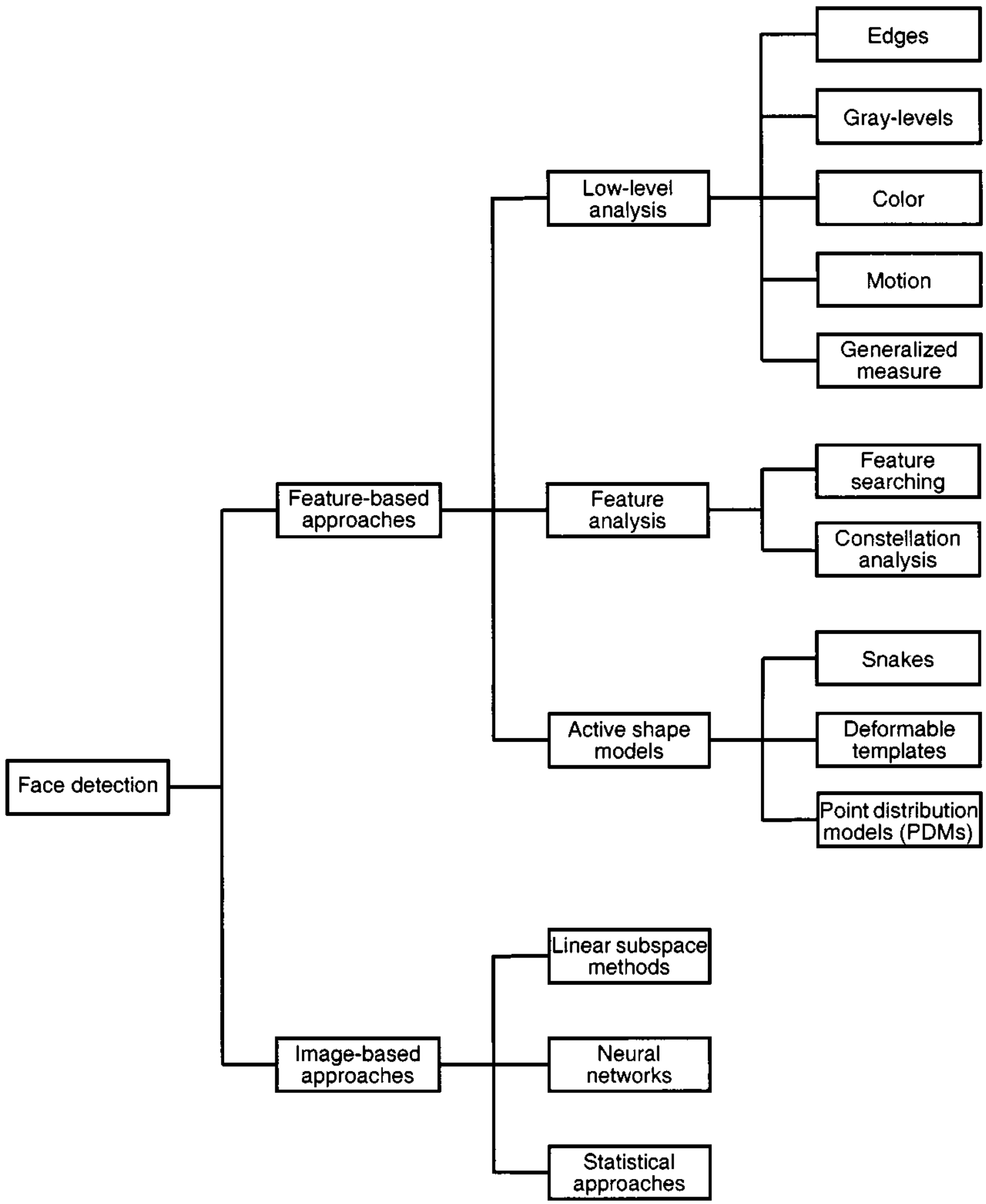


Figure. 2.1 Face detection divided into approaches.

CHAPTER III

CHALLENGES TO FACE DETECTION

Object detection is the problem of determining whether or not a sub-window of an image belongs to the set of images of an object of interest. Thus, anything that increases the complexity of the decision boundary for the set of images of the object will increase the difficulty of the problem, and possibly increase the number of errors the detector will make. Suppose we want to detect faces that are tilted in the image plane, in addition to upright faces. Adding tilted faces into the set of images we want to detect increases the set's variability, and may increase the complexity of the boundary of the set. Such complexity makes the detection problem harder. Note that it is possible that adding new images to the set of images of the object will make the decision boundary becomes simpler and easier to learn. One way to imagine this happening is that the decision boundary is smoothed by adding more images into the set. However, the conservative assumption is that increasing the variability of the set will make the decision boundary more complex, and thus make the detection problem harder.

There are many sources of variability in the object detection problem, and specifically in the problem of face detection [9]. These sources are outlined below.

3.1 Variation in the Image Plane

The simplest type of variability of images of a face can be expressed independently of the face itself, by rotating, translating, scaling, and mirroring its image. Also included in this category are changes in the overall brightness and contrast of the image, and occlusion by other objects. Examples of such variations are shown in **Figure 3.1.**

3.2 Pose Variation

Some aspects of the pose of a face are included in image plane variations, such as rotation and translation. Rotations of the face that are not in the image plane can have a larger impact on its appearance. Another source of variation is the distance of the face from the camera, changes in which can result in perspective distortion. Examples of such variations are shown in **Figure 3.1** in the following page.

The images of a face vary due to the relative camera-face pose (frontal, side, 45%, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.

Challenges in Face Detection

Figure 3.1 Examples of how face images vary between poses and between different individuals.

3.3 Facial Expression

The appearance of faces is directly affected by a person's facial expression as shown in **Figure 3.1.**

3.4 Shape Variation

Another source of variation is the shape of the object itself. For faces, this type of variation includes facial expressions, whether the mouth and eyes are open or closed, and the shape of the individual's face, as shown in some of the examples of **Figure 3.1.**

3.5 Occlusion

Faces may be partially occluded by other objects.

3.6 Image Orientation

Face images directly vary for different rotations about the camera's optical axis.

3.7 Lighting and Texture Variation:

Up to now, we have described variations due to the *position* and *orientation* of the object with respect to the camera. Now we come to variation caused by the object and its environment, specifically the object's surface properties and the light sources.

Changes in the light source in particular can radically change a face's appearance.

Examples of such variations are shown in **Figure 3.2** in the following page.

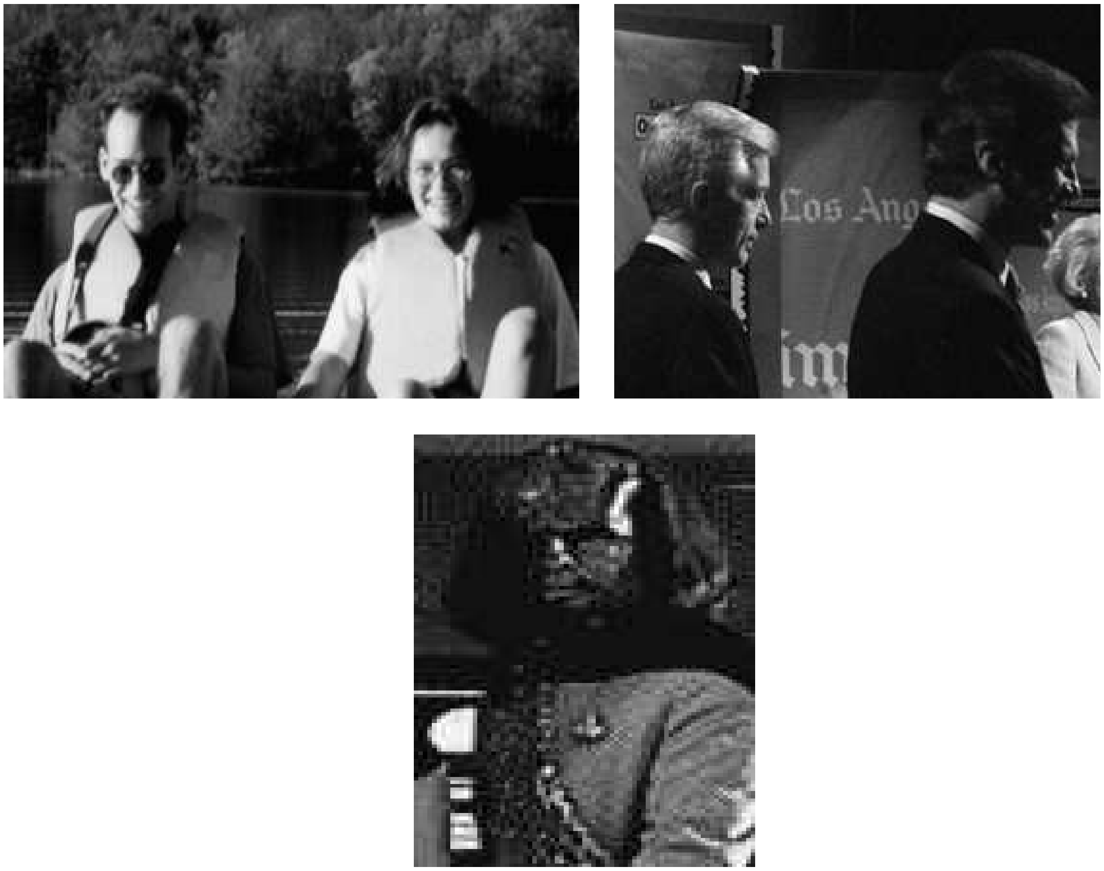


Figure 3.2 **Examples of how images of faces change under extreme lighting conditions.**

3.8 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.

3.9 Background Variation

In his thesis, Sung suggested that with current pattern recognition techniques, the view based approach to object detection is only applicable for objects that have “highly predictable image boundaries” [Sung, 1996]. When an object has a predictable shape, it is possible to extract a window which contains only pixels within the object, and to ignore the background. However, for profile faces, the border of the face itself is the most important feature, and its shape varies from person to person.

Thus the boundary is not predictable, so the background cannot be simply masked off and ignored. A variety of different backgrounds can be seen in the example images of **Figures 3.1** and **3.2**.

3.10 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.

CHAPTER IV

TERMINOLOGY

4.1 Binary Image

A binary image is an image that has only two possible values for each pixel. It is also familiar as bi-level or two level [10]. The names black and white , B & W. monochrome or monochromatic are used for this concept but this can also be used for grayscale image or one that have only one sample per pixel. In digital image processing, binary images often result from certain operations like “segmentations”, “thresholding” etc. it is formed by thresholding a grey scale or color image. The interpretation of the pixel’s binary values differs from device to device 0 is interpreted as black and 1 is interpreted as white.

4.2 Grayscale Image

A “grayscale or greyscale” digital image is an image in which the value of every pixel is a single sample [11]. In a gray scale image, red green and blue all have equal intensities in RGB. Thus, a single intensity value is used for every pixel. Image of this sort are typically composed of shades of gray varying from black at the weakest intensity to white at the strongest & with many shades of gray in between. Grayscale image are typically stored with & bits per sampled pixel which allows 256, (2^8=256) intensities.

i.e. for 8 bit grey scale image, there 256 are possible intensities.

4.2.1 Conversion of color image to grayscale image

The conversion of a color image to gray scale image is possible by first obtaining value of its red, green and blue (RGB) primaries.

Then it is necessary to add 30% red value + 59% green value + 11% blue. These percentages are chosen irrespective of the scales (0.0 - 1.0, 0 to 255, 0% to 100% etc).

The resultant level is the desired gray value. If we note the percentages it is clear that the percentage is higher for green and lower for blue. It is chosen in such a way due to the different relative sensitivity of normal human eye.

4.3 Pixel

A pixel (short for picture element) is a single point in a graphical image [12]. In our sample image there will be many pixels and each information element is an abstract sample. The intensity of each pixel is variable; in color system the dimension of variability of each pixel are red, green and blue or cyan, magenta, yellow and black.

A pixel is generally thought as the smallest complete sample of an image the greater are the chances for a close resemblance between the original and the result. However the number of pixels is sometimes called the resolution.

Pixels can be either rectangular or square. Each pixel in a monochrome image has a unique value, a correlate of perceptual brightness or physical intensity. Black is usually represented by the numeric value zero while white has the maximum value which is represented by 255 for en eight bit image; as 2^8=256 and the values range from o at minimum to 255 at maximum, while in a color image each pixel is usually represented as the re green and blue intensities.

4.4 RGB Color Model

The name of the model and the abbreviations 'RGB' come from the three primary colors; red green and blue [13]. This RGB model is an additive model where the three primary colors are combined in various ways and various proportions reproduce other colors the secondary colors in the visible spectrum. Levels of R.G and B can each range from o to 100 percent of full intensity. Each level is represented by the range of decimal numbers from 0 to 255, equivalent to range of binary numbers from 00000000 to 11111111 or hexadecimal 00 to FF.

4.5 Luminance

Luminance describes the amount of light that passes through or is emitted from a particular area and it falls within a given solid [14]. It is the photometric measure of the density of luminous intensity is a given direction. Luminance is the "emission" or "reflection" from flat surfaces.

When looking at a surface from particular angle of view, the amount of luminous power perceived by the eye will be indicated by the luminance. Thus the brightness of the surface is indicated by the luminance.

Luminance is defined by

$$\mathbf{Lv = d^2F / dAd\Omega Cos\theta} \qquad \textbf{(4.1)}$$

Where **Lv** – Luminance (cd/m^2)

F - Luminous Flux or Luminous Power (lm)

θ - Angle between the Surface Normal and the Specified Direction

A - Area of the Surface (m^2)

Ω - Solid Angle (Sr)

Relative luminance follows the photometric definitions of luminance, but with the values normalized to 1 or 100 for a reference white for4 color space such as XYZ, the litter Y refers to relative luminance for RGB color spaces however, relative luminance can be calculated from liner RGB components :

$$\mathbf{Y = 0.2126R + 0.7152G + 0.722\ B} \qquad \textbf{(4.2)}$$

Here, the co-efficient are all positive having a large green co-efficient and small blue co-efficient. The three forms the middle row of the RGB to XYZ color transformation matrix.

4.6 Segmentation

Segmentation is the process of portioning a digital image in to multiple images i.e. breaking the image in to sets pixels by doing [15]. So it is more meaningful and easier to analyze the picture. Image segmentation is used to locate objects and boundaries lines, curves etc) in images. A set of regions collectively cover the entire image when the image is segmented. Each of the pixels in the region are similar with respect to three characteristics color, intensity or texture. However, the adjacent regions are significantly different with respect to the three characteristics.

For our research, we have used the "*region-growing method*".

In a region growing method, a region is started with a single pixel adjacent regions are recursively examined and is added to the region if they are sufficiently similar to the region. A new region is started if the pixel is too dissimilar to the region. By doing so, a new image will be formed, having different regions.

We have used "8" neighborhood and "3" neighborhood for our thesis.

4.7 Covariance Matrix

In statistics and probability theory, the co-variance matrix's the matrix of covariance between elements of a vector [16]. The mean vector and covariance can be explained as follows:

Considering a matrix:

$$X = \begin{array}{ccc} \text{Length} & \text{Width} & \text{Height} \\ 4.0 & 2.0 & 0.60 \\ 4.2 & 2.1 & 0.59 \\ 3.9 & 2.0 & 0.58 \\ 4.3 & 2.1 & 0.62 \\ 4.1 & 2.2 & 0.63 \end{array}$$

Above, there are 5 sets of observations each sets measuring three variables- from left to right are *length, width* and *height* of certain object. (for example)

The 5 observation can be described by its mean vector and variance-covariance matrix.

Each row vector xi is another observation of the three variables. The mean vector consists of the means of each variable i.e. the mean of each column vector while the variance- covariance matrix consists fo the variance of the variables along the main diagonal and the covariance between each pair of variables in the other matrix position.

The covariance of the variables X and Y can be however computed as:

$$COV = \frac{\sum_{i=1}^{n} (X_i - \bar{x})(Y_i - \bar{y})}{n-1} \qquad \textbf{(4.3)}$$

where $\bar{x}$ and $\bar{y}$ are means of X and Y respectively.

The *mean vector* $\bar{x}$ is calculated and found out to be

$\bar{x}$ = 4.10 2.08 0.604;

(4.0 + 4.2 + 3.9 + 4.3 + 4.1)/5 = 4.10

(2.0 + 2.1 + 2.0 + 2.1 + 2.2)/5 = 2.08

(0.60 + 0.59 + 0.58 + 0.62 + 0.63)/5 = 0.604

and

the *variance-covariance matrix* **S** is calculated by

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})' \qquad \textbf{(4.4)}$$

Here n = 5;

And it is found out to be

$$S = \begin{matrix} 0.025 & 0.0075 & 0.00175 \\ 0.0075 & 0.0070 & 0.00135 \\ 0.00175 & 0.00135 & 0.00043 \end{matrix}$$

Thus, 0.025 is the variance of the length variable.

0.007 is the variance of the width variable.

0.00043 is the variance of the height variable.

And 0.0075 is the covariance between the length and width variable.

0.00175 is the covariance between the length and height variable.

0.00135 is the covariance between the width and height variable.

4.8 Thresholding

Thresholding is the simplest method of image segmentation [17]. It is a useful process to be able to separate out the regions of the image that corresponds to “objects” in which we are interested to work our from the regions of that image that correspond to background thresholding often provides an easy and convenient way for performing this segmentation on the basis of different intensities or colors in the foreground and background regions of an image. By thresholding it is possible to be able to see the areas of an image that consists of pixels whose value lien within a specified range, or band of intensities.

The input to a thresholding operation is typically a grayscale or color image. Its output is a binary image that represents the segmentation. In our paper, we represented the background with black pixels and the foreground with white pixels. For simple implementation, the segmentation is determined by a single parameter known as “intensity threshold” where each pixels of the image is compared with its threshold. While comparing, if the pixel's intensity is higher than the threshold then the pixel is set to, say, white in the output. If however, the pixel's intensity is lower than the threshold then the pixel is set to black. This is the simple implementation.

However, in more sophisticated implementation, multiple threshold can be specified such that a band of intensities can be set to white everything else is set to black. For color or multi-spectral images, it is possible to set different thresholds for each color channel and to set just those pixels within a specified cuboid in RGB space.

One another common variant is to set all the pixels corresponding to background as black, but leave foreground pixels at their original color/ intensity, such that no information is lost.

4.9 Adaptive Thresholding

As note earlier, thresholding is used to segment an image by setting all the pixels whose intensity values are above a threshold to a foreground value and setting all the remaining pixels to a background value [18]. Here a global threshold is used for all pixels and compared. But in an adaptive thresholding, the threshold dynamically changes over the image. This is a more sophisticated version of the thresholding and can accommodate changing lighting conditions in the image.

The lighting conditions that due strong illumination gradient or shadows.

Typically, adaptive thresholding takes a grayscale or color image as input and in the simplest implementation. It out puts a binary image representing the segmentation. It is used to separate desirable foreground image objects from background based on the difference in pixel intensities of each region. It does so by selecting an individual threshold for each pixel base on the range of intensity values in its local neighborhood. Thus it is possible to threshold an image whose global intensity histogram does not contain distinctive peaks.

4.10 Chroma Chart

A chroma chart is used to transform color image to skin likelihood image. This chart shows the likelihood of different colors representing the skin. The chroma chart will have two components representing the a and b values in the 1976 CIE LAB color co-ordinate system. It is used to transform a color image into a grayscale image in which the gray value will show the likelihood of the pixel belonging to the skin. But as some times, some regions are mistakenly considered as skin regions, discrimination of skin regions from non-skin region is possible by a process called skin segmentation. I.e. by segmenting the gray scale image it is possible to distinguish between skin region and non-skin region.

4.11 Connected Components Labeling

In Computer Science, Connected Components Labeling is a general approach to solve a number of common problems [19]. With a set of objects and connectivity information among them, it is possible to assign a unique label to each subset of objects, those that are connected to each other -- by connected component labeling. Each such subset of objects is a connected component or simply a component. The objects can be nodes of an abstract graph, variables in a Fortran common block or can be a pixel in an image (the one that we are using for our thesis).

Connected component labeling is commonly used in image analysis for computer vision and image understanding. In this case, a connected component-labeling algorithm is usually applied on a binary image with two types of pixels. These binary images are usually produced from another image processing step. There are two types of pixels in binary image, those are referred to as object pixels and background pixels. An object pixel (often represented with value 1) indicates something of interest. And these object pixel form objects to be labeled. The background pixels are typically assigned a fixed label and are treated separately from the object pixel. It is common to use integer labels for the computer, in which case, the value 1 is used to represent the foreground pixels (the object pixel) whereas the value 0 is used to represent the background pixels.

By using a two-dimensional (2D) array, a binary image can be represented in the simplest of all way.

The simplest approach for connected component labeling is to repeatedly scan the image to determine appropriate labeling until no further changes can be ma de to the assigned labels. Provisional label is the label assigned to an object pixel before assigning to the final assignment. For a 2D image, a forward scan assigns labels to pixels from left to right and top to bottom. Whereas a backward scan assigns labels to pixels from right to left and from bottom to top. Each time a pixel has been scanned; its neighbors that have been scanned are examined to determine an appropriate label to be assigned to the current pixel. The current pixel then receives a new provisional label if there is no object pixel in the neighborhood just scanned. However, if there are any object pixels in the neighborhood then the provisional labels of the neighbors are considered equivalent and a representative label is selected to represent all the equivalent labels, and thus the current pixel is assigned this representative label. This is how each equivalent labels are assigned with a label.

One simple strategy for selecting a representative is to use the smallest label. A more sophisticated labeling approach may have a separate data structure for storing the equivalence information or a different strategy to select a representative of the equivalent labels. The common algorithm for labeling images stored as 2D arrays can be divided into three groups. These groups are:

4.11.1 Multi-pass algorithm

The basic labeling algorithm described above is the best-known example of this group. But the shortcoming of this algorithm is that the number of scans can be large. In order to control the number of iterations, the direction of scans can be altered, or the equivalence information can be directly manipulated.

4.11.2 Two-pass algorithm

Many algorithms in this group operate in three distinct phases.

1. Scanning phase
2. Analysis phase - This phase analyzes the label equivalence information to determine the final labels.
3. Labeling phase - This phase assigns the final labels to the object pixels by passing through the image a second time. One of the most efficient data structures for representing the equivalence information is the union-find data structure.

4.11.3 One-pass algorithm

In this group, an algorithm scans the image to find the unlabeled object pixel and then assigns the same labels to all the connected object pixels. The most efficient algorithm in this group is the Contour Tracing algorithm by Chang et al.[] Algorithms in this group need to go through the image only once, typically with an irregular access pattern. For example, each time an unlabeled object pixel is found, the Contour Tracing algorithm follows the boundary of the connected component until it returns to the starting position. It then fills in the labels for the object pixels in the interior of the component.

In most of the cases the images are not stored as simple 2D arrays; therefore it is not possible to use the above algorithms directly. However, most of the algorithms for the above complex image formats are built upon the algorithms from groups 1 and 2.

CHAPTER V

METHOD IMPLEMENTED

The current evolution of computer technologies has envisaged an advanced machinery world, where human life is enhanced by artificial intelligence. Indeed, this trend has already prompted an active development in machine intelligence. **Computer vision, for example, aims to duplicate human vision.** Traditionally, computer vision systems have been used in specific tasks such as performing tedious and repetitive visual tasks of assembly line inspection. Current development in this area is moving toward more generalized vision applications such as face recognition and video coding techniques.

Figure. 5.1 Typical training images for face detection.

Many of the current face recognition techniques assume the availability of frontal faces of similar sizes [20, 21]. In reality, this assumption may not hold due to the varied nature of face appearance and environment conditions. Consider the pictures in **Figure. 5.1** These pictures are typical test images used in face classification research. The exclusion of the background in these images is necessary for reliable face classification techniques.

However, in realistic application scenarios such as the example in **Figure. 5.2**, a face could occur in a complex background and in many different positions. Recognition systems that are based on standard face images are likely to mistake some areas of the background as a face. In order to rectify the problem, a visual front-end processor is needed to localize and extract the face region from the background.

Face detection is one of the visual tasks which humans can do effortlessly. However, in computer vision terms, this task is not easy. A general statement of the problem can be defined as follows: Given a still or video image, detect and localize an unknown number (if any) of faces. The solution to the problem involves segmentation, extraction, and verification of faces and possibly facial features from an uncontrolled background. As a visual front end processor, a face detection system should also be able to achieve the task regardless of illumination, orientation, and camera distance. Our thesis aims to provide insight into the contemporary research of face detection in a structural manner. Chellappa *et al.* [22] have conducted a detailed survey on face recognition research. In their survey, several issues, including segmentation and feature extraction, related to face recognition have been reviewed. One of the conclusions from Chellappa *et al.* was that the face detection problem has received surprisingly little attention. This has certainly changed over the past five years.

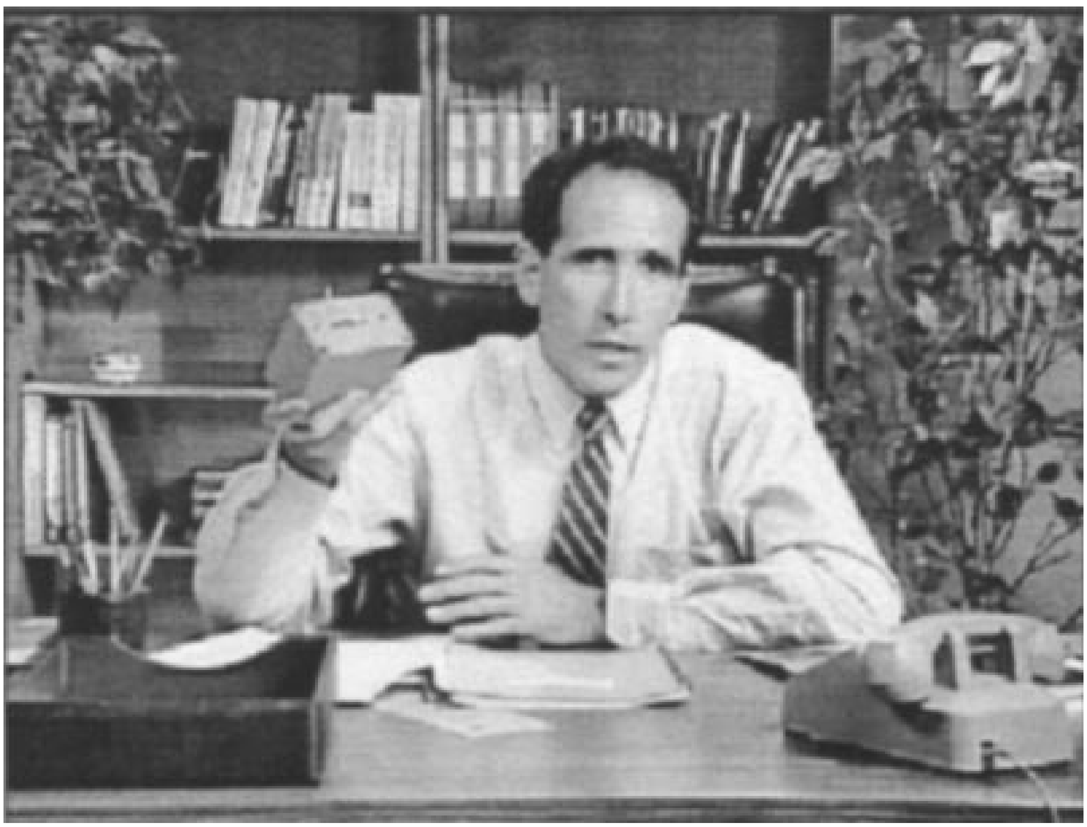


Figure. 5.2 A realistic face detection scenario.

In computer vision, face detection is one of the major research areas. Detection of human faces is becoming a very important task in various applications, such as video surveillance and security control system, intelligent human computer interface, face recognition, content-based image retrieval, multimedia applications on the web like video conferencing and face database management. It is presumed in most of the current face recognition systems that faces are readily available for processing. But in reality, the task is not as easy as it seems because images contains not just faces but also other objects in the background (as is shown in figure 5.2).

Therefore, it is important to design a system that will detect, locate and segment faces in images, such that these faces can be given as input to face recognition system. Face detection being so trivial for human beings still remains a challenging and difficult problem to enable a computer to do so. There are however various reasons that leads to this difficulty.

Because of the diversity of variations due to external factors such as:

- Lighting conditions
- Human race
- Contrast between face (foreground) and background
- Orientation of the face
- Illumination and

Because of the fact that human face changes with respect to internal factor such as:

- Facial expression
- Spectacles
- Beard
- Mustache etc.

Various approaches to face detection have been developed as in [23] and [24]. These techniques utilize various techniques such as

- Principle Component Analysis,
- Neural Networks,
- Machinery Learning,
- Information Theory,
- Geometric Modeling,
- Template Matching,
- Hough Transform,
- Motion Extraction,
- Color Analysis and many more.

However, a large numbers of face and non-face training samples are required for the neural network based approaches [25], [26] and the view based process [27] and they are designed primarily to locate frontal in grayscale images. Schneiderman and Kanade [28] extended their learning based approach for the detection of frontal faces to profile views. Face detection based on Markov random fields [29] and Markov Chains [30], make use of the spatial arrangement of pixel gray values. A feature based approach that uses geometrical facial features with belief networks [31] provides face detection for normal frontal views. Geometrical facial templates and the Hough transform were incorporated to detect grayscale frontal faces in real time applications.[32].

In our thesis, we implemented an approach to face detection based on ***skin color***. We have developed and implemented a color-based technique for detecting frontal human faces in which they appear. Using this technique of detecting human face has many advantages. Color processing is much faster than processing other facial features. Using this technique includes no classification method like training neural networks.
In addition, under certain lighting conditions, color is orientation invariant.

For this method we have used *two image processing steps*.

First, skin regions are separated from non-skin regions with the help of color information and simple threshold that shows the likelihood of skin colors which is used to generate a gray scale image from the original color image.

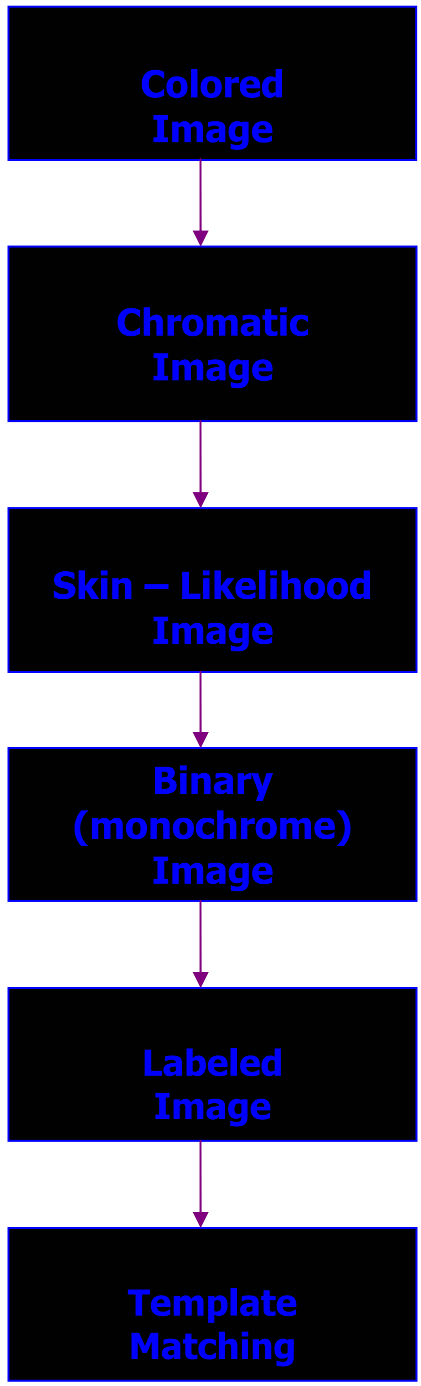
Second, the frontal human face within the skin region is located. The image has the property that the gray values at a pixel show the likelihood of the pixel representing the skin. The gray scale image is segmented to separate the skin regions from the non-skin regions. The luminance component is used, together with template matching to determine if a given skin region is frontal human face or not.

Our project is divided in several sections, each section describing a part of the process for face detection.

The thesis is implemented in MATLAB using the MATLAB Image Processing Toolkit. We have used **MATLAB 7.1** as our Programming language.

A flowchart showing the entire process starting from the basic (color image) to the point where the face is detected is shown in **Figure 5.1**:

5.1 Flowchart



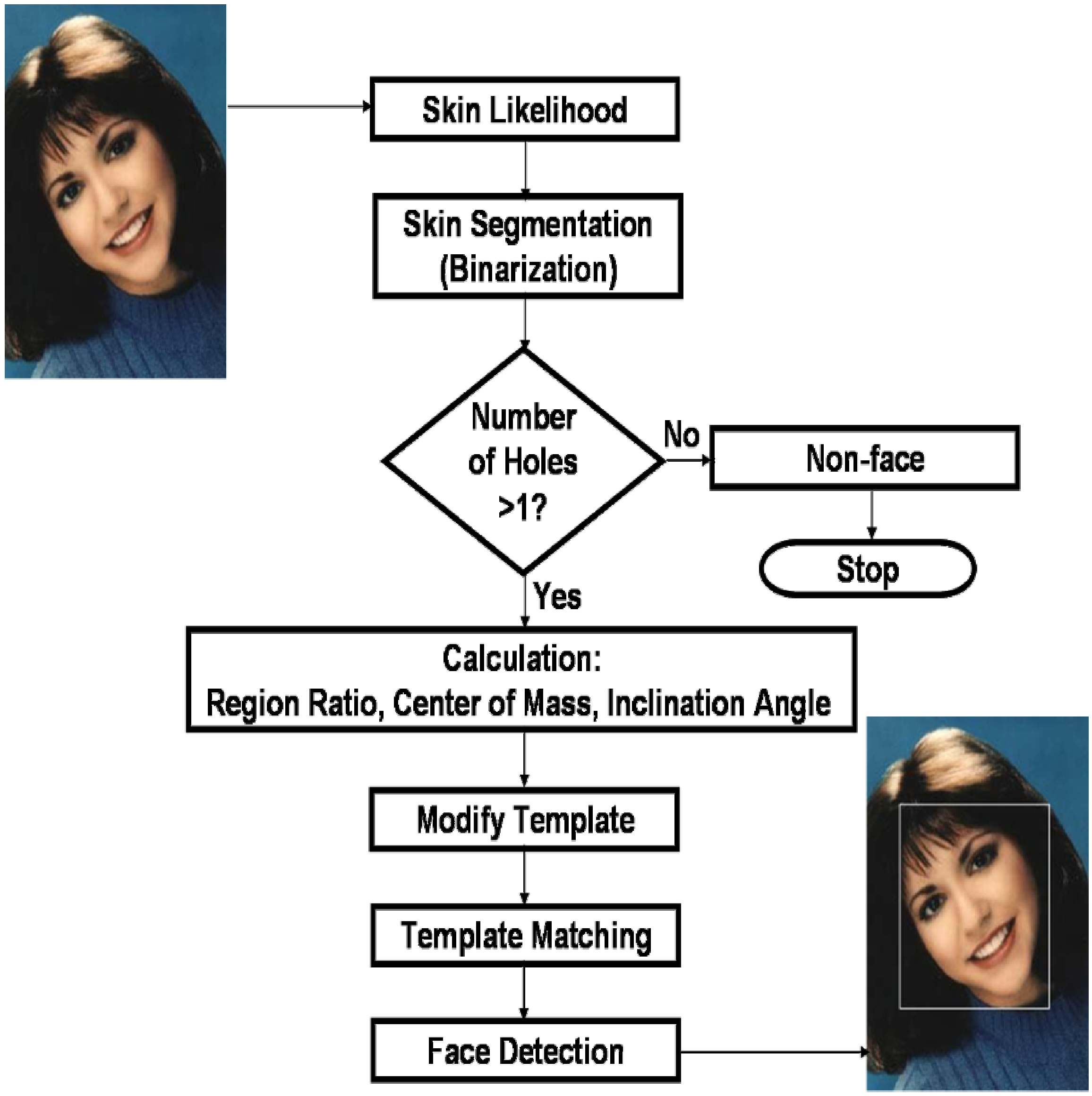


Figure 5.3 The Proposed Face Detection Algorithm

CHAPTER VI

PROCEDURE

6.1 Skin Color Model

A reliable skin color model that is adaptable to people of different skin colors and different lighting conditions is necessary if we want to segment skin regions from non-skin regions [33]. In this section, we describe a model of skin color in the chromatic space for skin segmentation. [34]

The common RGB representation of color images is not suitable for characterizing skin-color. In the RGB space, the triple component (r, g, b) does not only represents color but it also represents *luminance*. Luminance may vary across a person's face due to the ambient lighting and it is not a reliable measure in separating skin from non-skin region [35]. Therefore, we have to remove luminance. It is possible to remove luminance from the color representation in the chromatic color space. Chromatic colors [36], also known as "pure" colors in the absence of luminance, are defined by a normalization process shown below:

$$r = R/(R+G+B)$$

$$b = B/(R+G+B) \qquad \textbf{(6.1)}$$

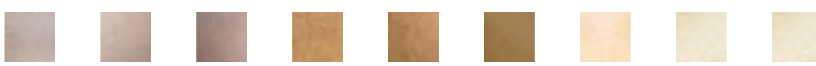
If we notice, then we can see that the component g is avoided. That is done in order to remove luminance.

For our thesis, we have taken a total of 27 skin samples. Our skin samples were taken from persons of different ethnicities:

- Asian
- Caucasian
- African

Each of the skin samples are shown in **Figure 6.1** below.

9 Asian skin samples



9 African skin samples

9 Caucasian skin samples

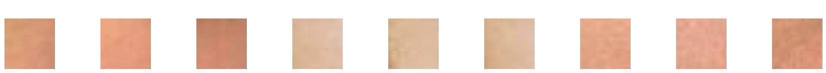


Figure 6.1 Skin samples

For each skin samples shown above, the red mean and the blue mean are extracted using equation **6.!**. After obtaining the red mean and blue mean of 27 skin samples, the overall mean (for both red and blue) is calculated.

In the next section, we discuss the process of transforming a color image to skin-likelihood image.

6.2 Skin Likelihood Image

In the last section, we have converted a color image to a chromatic image. Therefore, if a pixel, having transform from RGB color space to chromatic color space, has a chromatic pair value of (r, b), the likelihood of skin for this pixel can be computed as follows:

$$Likelihood = \exp[-0.5(x-m)^T C^{-1}(x-m)] \qquad \textbf{(6.2)}$$

Here,

x = (red value, blue value) (of chromatic image)

m = (mean red value, mean blue value)

C = covariance of chromatic image

Beginning with a color image, the first stage is to transform it to a skin-likelihood image. This process involves transforming every pixel from RGB representation (color image) to chroma representation (chromatic image) and determining the likelihood value based on the equation given above. The skin-likelihood image will be a gray-scale image whose gray values represent the likelihood of the pixel belonging to skin. A sample color image and its resulting skin-likelihood image is shown in **Figure 6.2.1**. The skin region (like the face) is shown brighter than the non-skin regions.

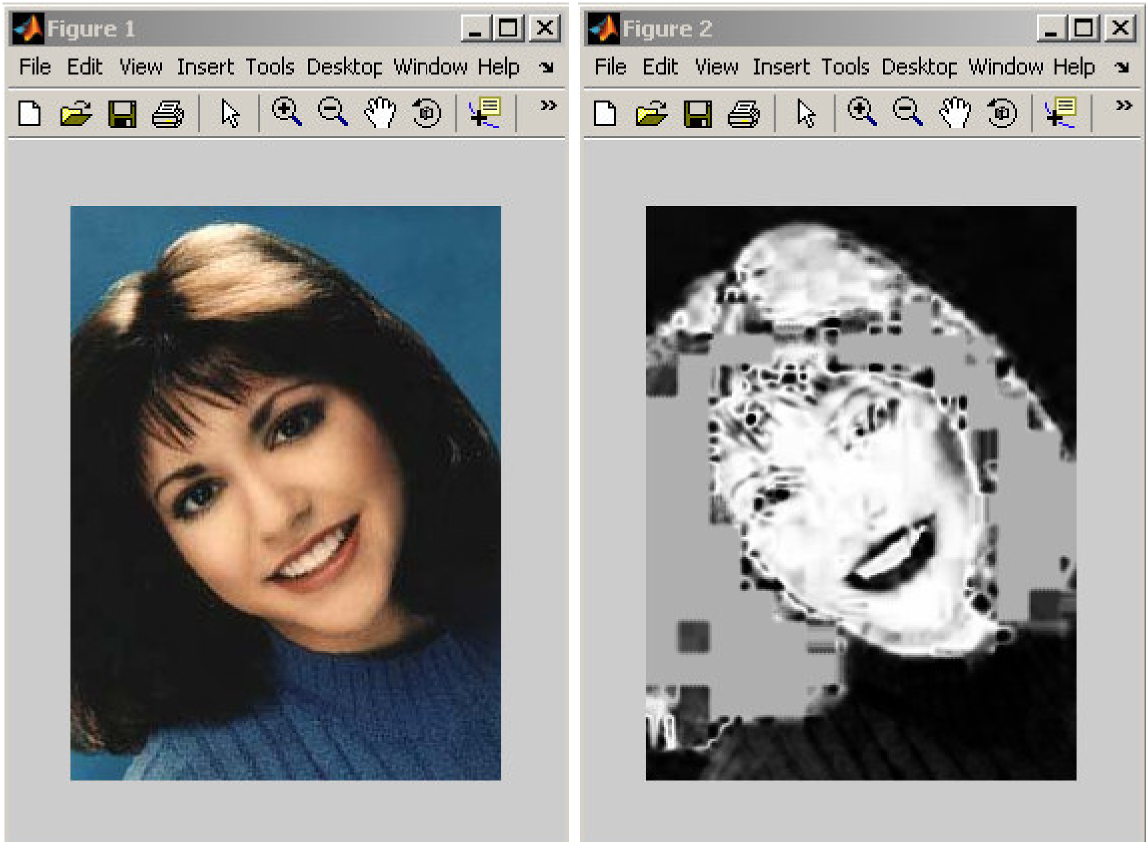


Figure 6.2.1 (Left) The Original Color Image (Right) The Skin-Likelihood Image

However, one important point to consider is that the detected regions may not necessarily correspond to skin. If we look at **Figure 6.2.1**, a part of the hair has been mistakenly considered as a skin region. This has occurred as the lady's skin color matches with part of her hair. Therefore, that particular part of her hair has been considered as skin region. It is only reliable to conclude that the detected regions have the same color as that of the skin. The important point here is that this process can reliably point out regions that do not have the color of the skin and such regions would not need to be considered anymore in the face finding process. Therefore, those regions that have a black portion are not the skin regions and we can ignore those regions and work with the white regions only.

However in the white region, non-skin regions that has been mistakenly considered as skin regions needs to be fixed. And this can be done by a process called binarization and labeling.

In the next section, we discuss the process of converting a skin-likelihood image to the binary image.

6.3 Binary (monochrome) Image

Since the skin regions are brighter than the other parts of the image, the skin regions can be segmented from the rest of the image through a process called thresholding. To process different images of different people with different skin color, we use a global thresholding. We set our global thresholding value by experiment. After analyzing 15 to 25 images, we set our global thresholding value to be 195. With this experimented value we do a comparison. We go to the skin likelihood image and then compare the experimented value with the value at each pixel of the skin likelihood image.

If the value of the pixel is greater than or equal to the experimented value then we make the pixel *white*. This means that it is a skin region.

And if the experimented value is less than the value at the pixel then we make the pixel *black*, which means that it is a non-skin region.

Using this technique of thresholding, many images yield good results; the skin-colored regions are effectively segmented from the non-skin colored regions. The binary image of the previous color image resulting from this technique is shown in **Figure 6.3.1**

It is observable from Figure 6.3.1 that not all detected skin regions contain faces. Some correspond to other exposed part of the body (which are not skin region), while some corresponds to objects with colors similar to those of the skin, like the hair (which is not a skin region as well). Hence the second stage of face finder will employ facial features to locate the face in all these skin-like segments.

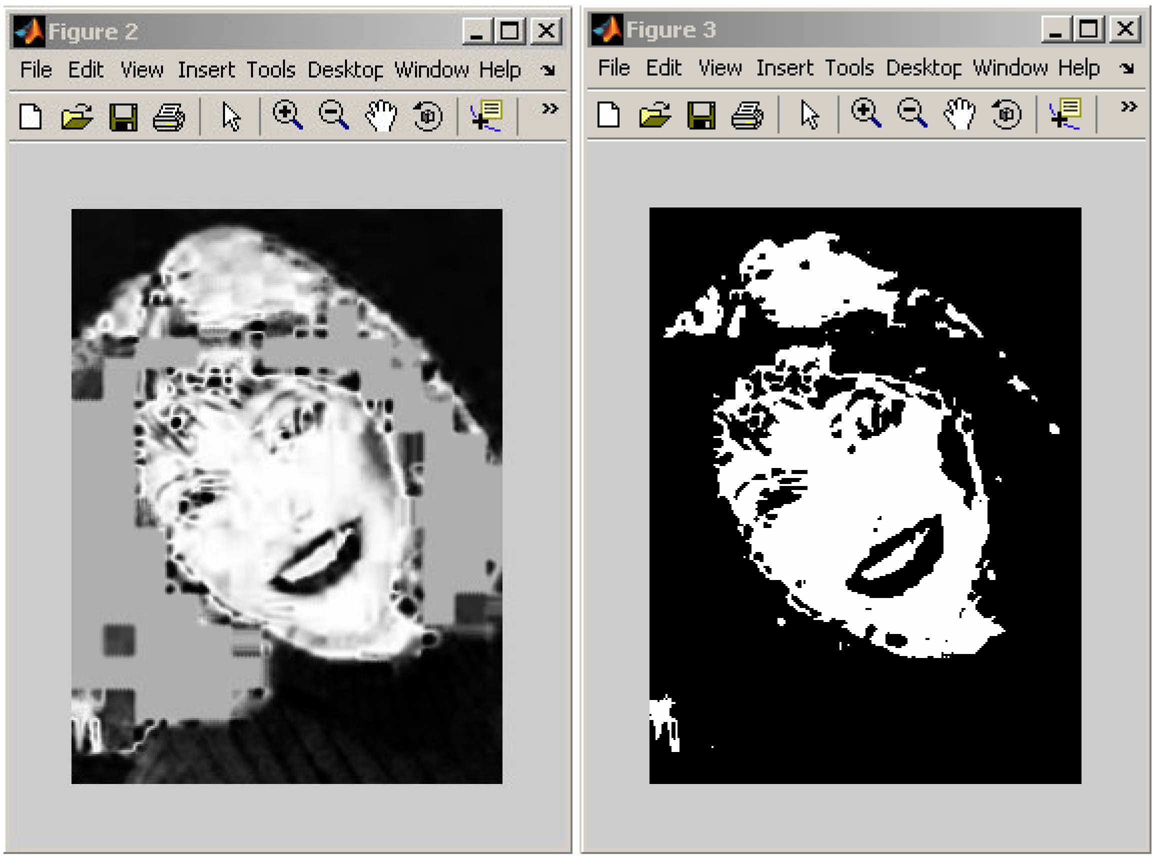


Figure 6.3.1 (Left) The Skin-Likelihood Image (Right) The Binary Image

In the next section we discuss the process of conversion o a binary image to its labeled image.

6.4 Labeled Image

Using the result from the previous section, we proceed to determine which regions can possibly determine a frontal human face. To do so, we need to determine the number of *skin regions* in the image.

A **skin region** is defined as a closed region in the image, which can have 0, 1 or more holes inside it. Its color boundary is represented by pixels with value 1 for binary images. We can also think about it as a set of connected components within an image. All holes in a binary image have pixel value of zero (black).

The process of determining how many connected regions we have in a binary image is by *labeling*. A label is an integer value. Labeling of an image is done in order to identify each connected regions. We have used the MATLAB built in function to generate the labeling of a pixel. If any of the neighbors had a label i.e. an integer value, then we label the current pixel with that value. If not, then we use a new integer value.

Now, if we observe the binary image of **Figure 6.3.3** we will notice that there are many connected regions. But only all connected regions correspond to face region. There are a lot of unnecessary region that do not correspond a face. So we need to eliminate those regions. And it is done by a process called *Short-Listing*. Short-Listing is done to reduce the number of connected regions. To do so, we consider a certain area for a face. And then we compare. We compare each connected region with the experimented area. If the area of the connected region area is less than the experimented “area”, then we make the region black. This means it is not a face region. And if the connected region area is equal to the experimented area then we make that region white. This means that this region is the face region. This is how labeling and then short listing is done.

The conversion of the binary image to the labeled image is shown in **Figure 6.4.1**

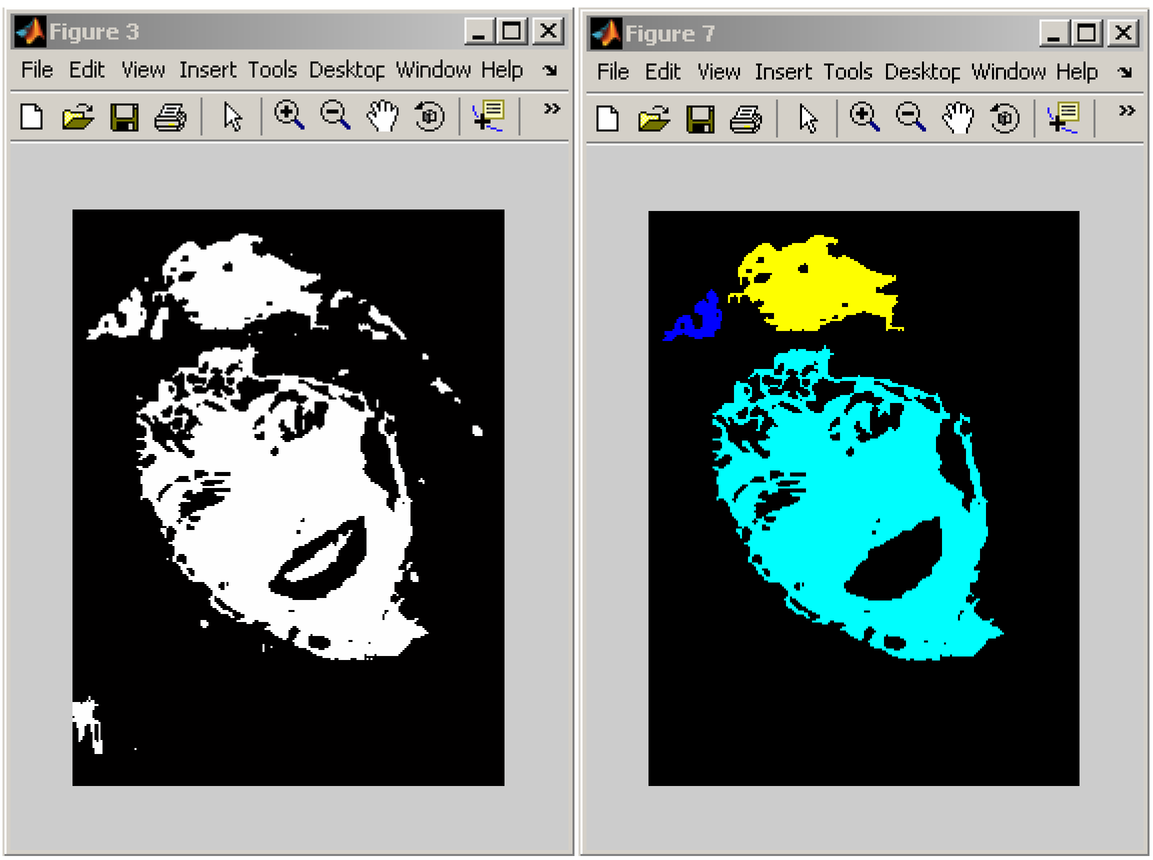


Figure 6.4.1 (Left) The Binary Image (Right) The Short-Listed Image

If we compare between both the pictures then we can see that in the second image a lot of connected regions has been eliminated. In the binary image, there are a lot of connected regions. But by labeling and by the process short-listing, it has been possible to eliminate any unnecessary region that we do not need.

In the next section we describe how we process the candidate faces.

6.5 Processing the Face

For processing the face, we find the regions, which ***may*** be a face. It is observable from the Short-listed image of Figure 6.4.1 that there are three connected regions in the labeled image. But not all connected regions correspond to a face. This is why we find the regions that *may* be a face. Then we find the *probable candidate* for face. The candidates are obtained from the labeled image. For example from the labeled image in Figure 6.4.1, it is possible to obtain a candidate for face. Then each of the candidates obtained are processed at a time.

6.5.1 Number of Holes inside a Region

After experimenting with several images, we come to a decision that a skin region should have at *least* **one** hole inside that region. Therefore, we get rid of those regions that have no holes. To determine the number of holes inside a region, we compute the **Euler number** of the region. It is defined as follows:

$$\mathbf{E = C - H} \qquad \textbf{(6.3)}$$

where E: is the Euler number

C: The number of connected components

H: The number of holes in a region.

Therefore, the Euler number is the difference between *number of connected components* and *holes in a region*.

For our thesis, we already set the number of connected components, C (i.e. the skin region) to 1 since we are considering 1 skin region at a time.

Therefore, the number of holes is:

$$\mathbf{H = 1 - E} \tag{6.4}$$

And, number of holes = 1 – (Euler number).

where H: The number of holes in a region

E: The Euler number.

After the system has determined that a skin region has more than one hole inside the region, we proceed to analyze some characteristics in that particular region. We first create a new image with that particular region only. And we set the rest is set to black.

6.5.2 Center of Mass

Now, in order to study the region, it is necessary to first determine its *area* and its *center* of the region. This can be done in many ways. However, one efficient way is to compute the *center of mass* (i.e., centroid) of the region [37]. The center of area in binary images is the same as the center of the mass and it is computed as shown below:

$$\bar{x} = \frac{1}{A}\sum_{i=1}^{n}\sum_{j=1}^{m} jB[i,j]$$

$$\bar{y} = \frac{1}{A}\sum_{i=1}^{n}\sum_{j=1}^{m} iB[i,j] \qquad \textbf{(6.5)}$$

where: B is the matrix of size [n x m] representation of the region. A is the area in pixels of the region

One point to note is that for the computation above, we are also considering the holes that the region has.

In **Figure 6.5.1** we have included an image to show the Center of Mass as Axes

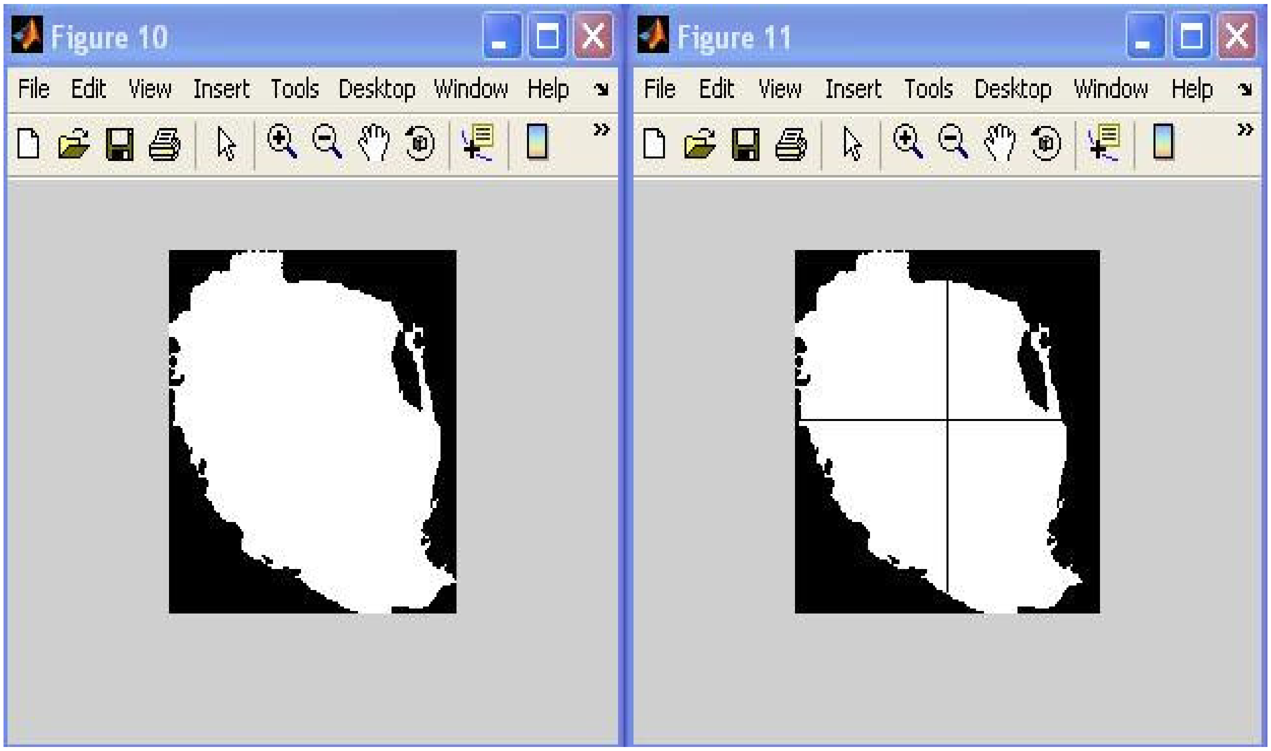


Figure 6.5.1 (Left) The face region without holes (Right) Center of Mass of the region

6.5.3 Orientation

Most of the faces we considered in our thesis are vertically oriented. However, some of them might have a little inclination as is shown in Figure 6.5.2.

Faces with certain degree of inclination.

Figure 6.5.2. Tilted Face

Therefore for a tilted face of this sort, we will have to rotate the face. One way to determine a unique orientation is by elongating the object. The orientation of the axis of elongation will determine the orientation of the region. In this axis we will find that the inertia should be the minimum.

The axis will be computed by finding the line for which the sum of the squared distances between region points and the line is minimum. In other words, we compute the least-squares of a line to the region points in the image [38]. At the end of the process, the angle of inclination (theta) is given by:

$$\theta = \frac{1}{2}\arctan\frac{b}{a-c} \qquad \textbf{(6.6)}$$

$$a = \sum_{i=1}^{n}\sum_{j=1}^{m}(x'_{ij})^2 B[i,j] \qquad b = 2\sum_{i=1}^{n}\sum_{j=1}^{m}x'_{ij}y'_{ij}B[i,j] \qquad c = \sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij})^2 B[i,j] \tag{6.7}$$

$$\begin{aligned} x' &= x - \bar{x} \\ y' &= y - \bar{y} \end{aligned} \tag{6.8}$$

By using the equation above, we can successfully find out the **angle of inclination**.

6.5.4 Height, Width and Region Ratio

Until now we have the *center of the region* and its *inclination* i.e. the angle of the inclination. But we need to determine one more thing. We need to determine the *width* and *height* of the region. With the width and height of the region, we will be able to resize our template face so it has the same width and height of our region.

In order to do so, first, we fill out the candidate's holes that the region might have. This is to avoid problems when we encounter holes. Since the image is rotated some angle theta (the angle of inclination), the need to rotate our region -theta degrees so that it is completely vertical as shown in **Figure 6.5.1**.

Rotation is required to find height, width and angle of inclination.

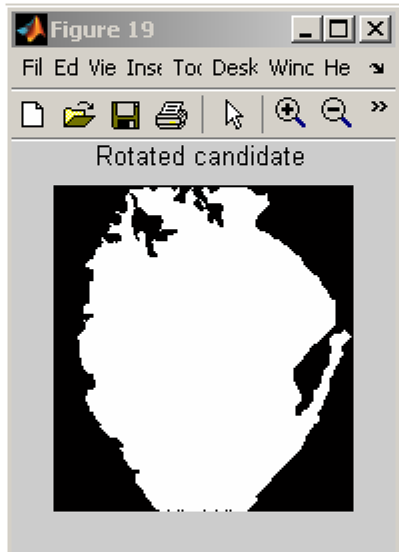


Figure 6.5.3 Rotated Candidate

We now proceed in order to determine the height and width by moving *4 pointers*: one from the **left**, **right**, **top** and **bottom** of the image. If we find a pixel value different from 0 i.e. a white region, we stop and this is the coordinate of a boundary. When we have the 4 values, we compute the height by subtracting the bottom value and top value; height = (bottom – top) values and we compute the width by subtracting the right value and the left value; width = (right - left) values. The Region ratio is then calculated. It is calculated as follows:

Region Ratio = Ratio of height to width = height/width **(6.9)**

The width and the height of the region can be used to improve our decision process. The height to width ratio of the human faces is around 1. So in order to have less misses we determined that a minimum good value should be 0.8. Ratio values that are below 0.8 do not suggest a face since human faces are oriented vertically.

In addition, the ratio should also have an upper limit. We determined the upper limit by analyzing the results in our experiments. We found that a good upper limit should be around 2.1. There are some situations however, that we indeed have a human face, but the ratio is higher. This happens when the person has no shirt or is dressed in such a way that part of the neck and below is uncovered. In order to account for these cases, we set the ratio to be 1.6 and eliminate the region below the corresponding height to this ratio.

6.6 Template Face

One of the most important characteristics of our method is that it uses a *human face template* to take the final decision of determining if a skin region represents a face or not. Therefore, with the template face we can find which of the candidate is a 'face'. This template is however chosen by averaging 16 frontal view faces of males and females wearing no glasses and having no facial hair. The template that we used is shown in **Figure 6.6.1** below**.**

If we observe Figure 6.6.1 we see that the left and right borders of the template are located at the center of the left and right ears of the averaged faces. In addition, the template is vertically centered at the tip of the nose of the model.

The template face to be used to find which of the candidates is a face.

Figure 6.6.1. Template face (model) used to verify the existence of faces in skin regions.

At this point, we have all the required parameters to do the matching between the part of the image corresponding to the skin region and the template human face. Therefore with the part of the image corresponding to skin region and template human face, template matching is possible.

Template matching is described in the next section.

6.7 Template Matching

In this section we show how to the matching between the part of the image corresponding to the skin region and the template face is done.

But before doing so, it is necessary to *resize* and *rotate* the template face.

6.7.1 Resizing and Rotating

The template face is resized according to the height and width of the regions computed in 6.5.4.

And the resized template face is rotated according to –theta i.e. according to the angle of inclination, such that the template face is aligned in the same direction the skin region is.

The resized and rotated template face looks as is shown in **Figure 6.7.1**

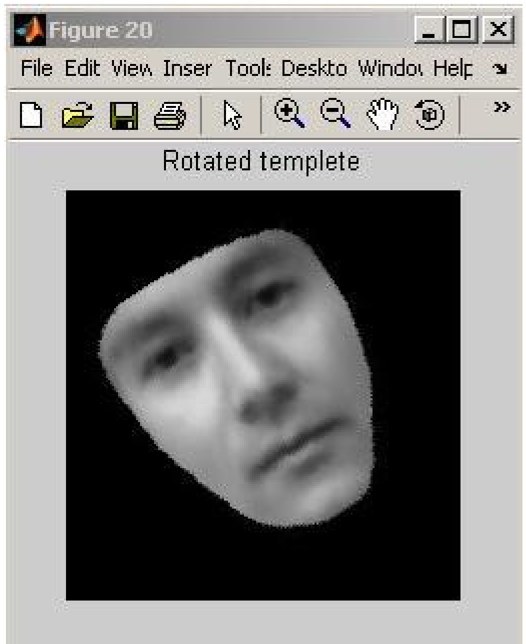
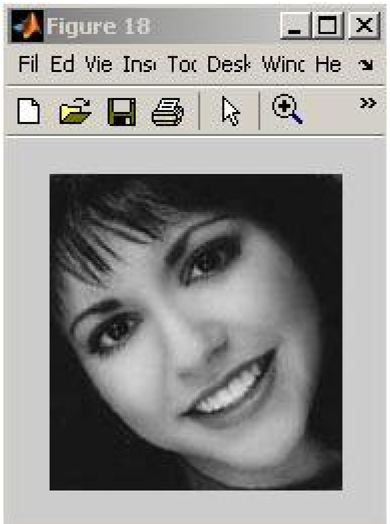


Figure 6.7.1 Resized – Rotated template face

The figure above shows the template face after it has been resized and rotated. If however, we look at it closely, then we will notice that the template face is the same size as the original image (color image) and it is rotated in the same angle (-theta) the original image is.

The comparison between both the images is shown in **Figure 6.7.2.**



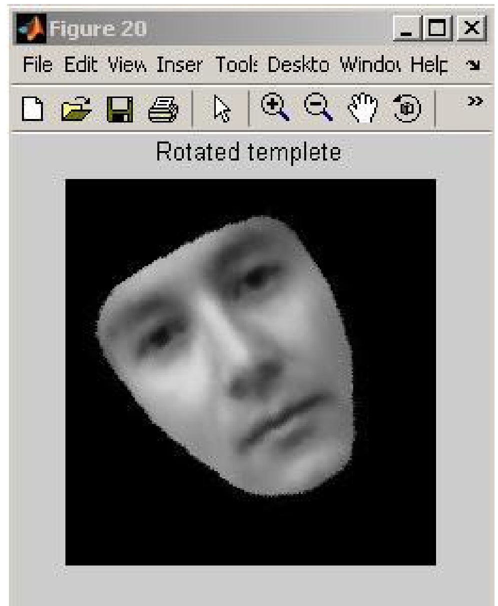


Figure 6.7.2 (Left) "Face" candidate (Right) Resized – Rotated template face

6.7.2 Cropping

The rotation process usually makes the image bigger, i.e. it adds black pixels to the image. The resized and rotated template therefore contains *extra* area. If we look at Figure 6.7.2 and notice the template face, then we will see that the 'face' is the same size and rotated in the same angle as the original image. But the difference is that there are some black regions that have been added. Therefore, this extra black pixel has to be removed and it is removed by cropping.

We now compare the cropped template image with the candidate face. This comparison is shown in **Figure 6.7.3**.

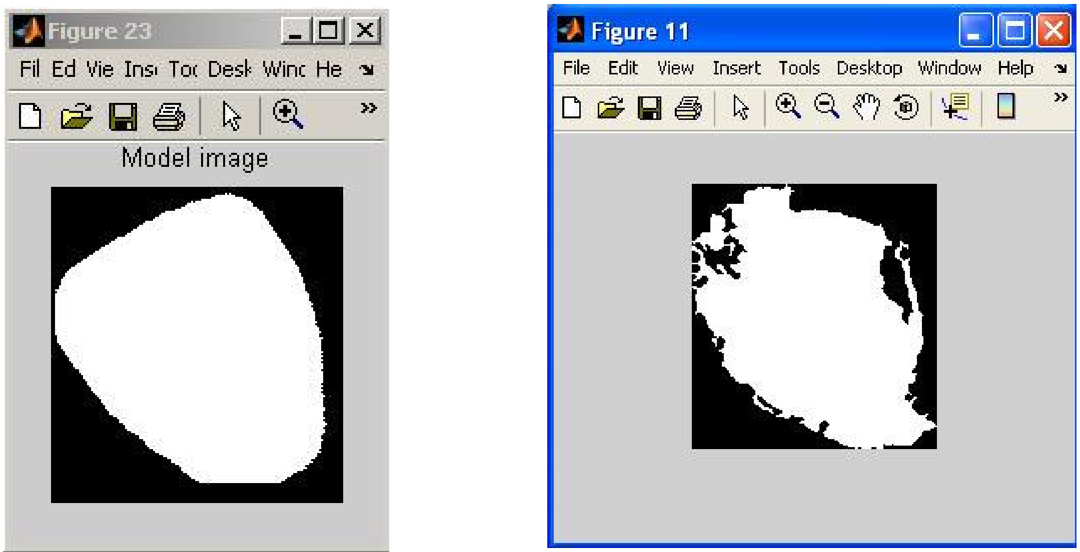


Figure 6.7.3 (Left) Resized – Rotated template face (Right) Face region

We then compute the *cross-correlation value* between the part of the image corresponding to the skin region and the template face properly processed and centered. The cross correlation value is taken to be greater than 0.45. Then a comparison is carried out.

If the above (experimented) value is and within the range of the region ratio, (the value of the region ratio is taken between 0.8 and 2.1 as discussed in the previous section.) then the region is a face. If not, then the region is not a face.

Finally, we get the coordinates of the part of the image that has the template face. With these co ordinates it is possible to draw a rectangle in the original color image. We draw a rectangle in the original color image and this is the output of the system which in our case, detected the face of the lady. The output is shown in **Figure 6.7.4**.

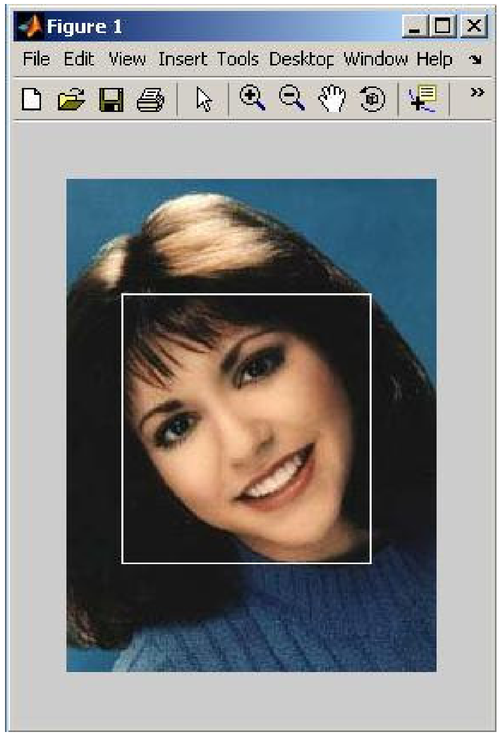


Figure 6.7.4 Final Result
Probable face displayed in the box

CHAPTER VII

RESULTS AND DISCUSSION

Following are some of the results which were created by the program. In total, there will be five for each image plus one more if any detection occurs. In other words, for any detection, there will be six images and five otherwise.

Detected images can be broadly categorized into three types: **True Detections**, **False Positives** and **False Negatives**. The definitions of each of these terms are given in the respective sections.

True Detections

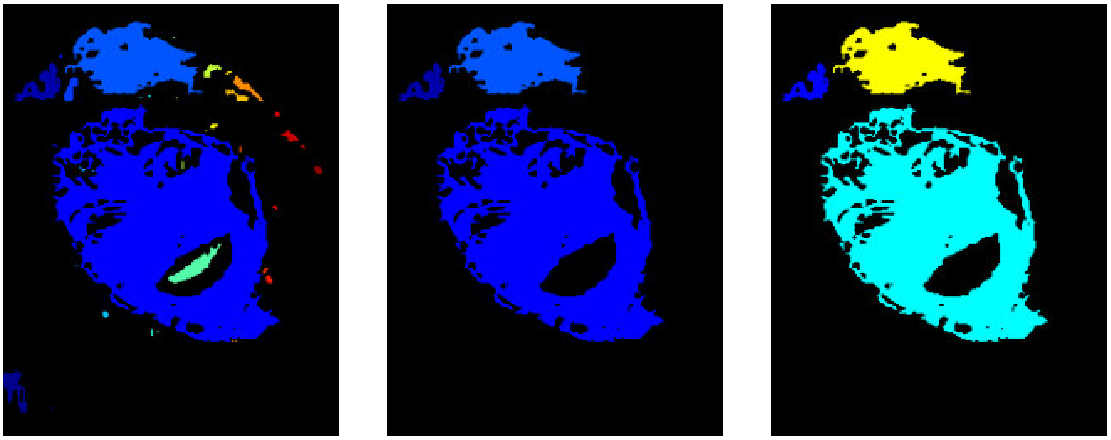
True detections mean that the detected region contains a face. Here are some of their results.

Image: image6.jpg

Figure 7.1.1: Original Image

Results:

Figures 7.1.2 & 7.1.3: Skin-likelihood Image & Binary Image



Figures 7.1.4, 7.1.5 & 7.1.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

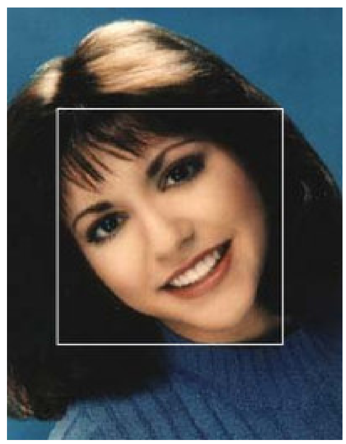


Figure 7.1.7: Detected Image

Discussion:

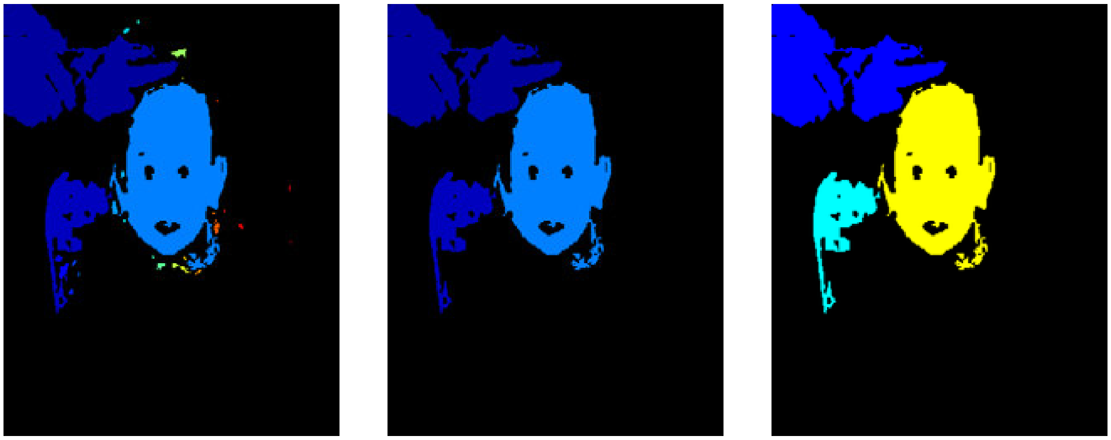
As can be seen from Figure 7.1.7, the face region has been successfully detected by the white box.

Image: asian1.jpg

Figure 7.2.1: Original Image

Results:

Figures 7.2.2 & 7.2.3: Skin-likelihood Image & Binary Image



Figures 7.2.4, 7.2.5 & 7.2.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.2.7: Detected Image

Discussion:

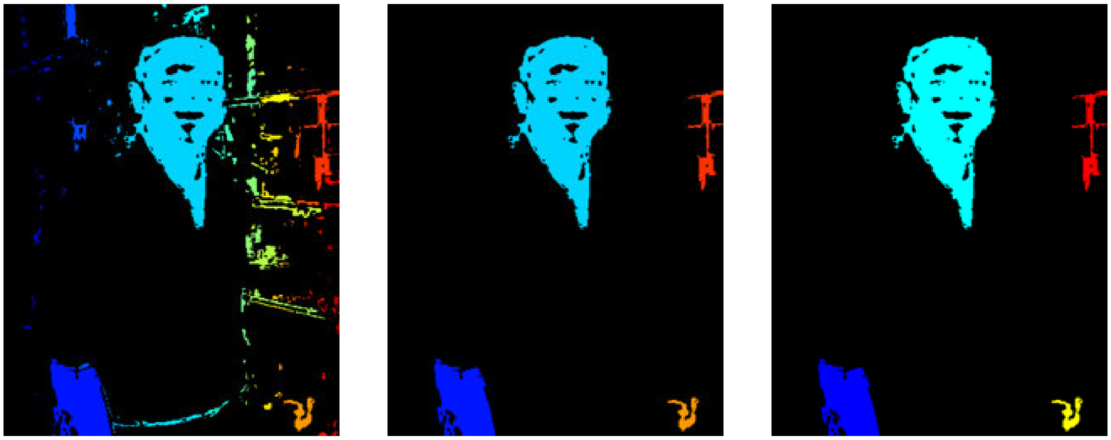
The result is very similar to that of Figure 7.1.7, as the facial region is detected by the white box. There is, however, the black clothing of the baby included in the detection. But in overall, the region does contain the face. The hands were not detected.

Image: image1.jpg

Figure 7.3.1: Original Image

Results:

Figures 7.3.2 & 7.3.3: Skin-likelihood Image & Binary Image



Figures 7.3.4, 7.3.5 & 7.3.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.3.7: Detected Image

Discussion:

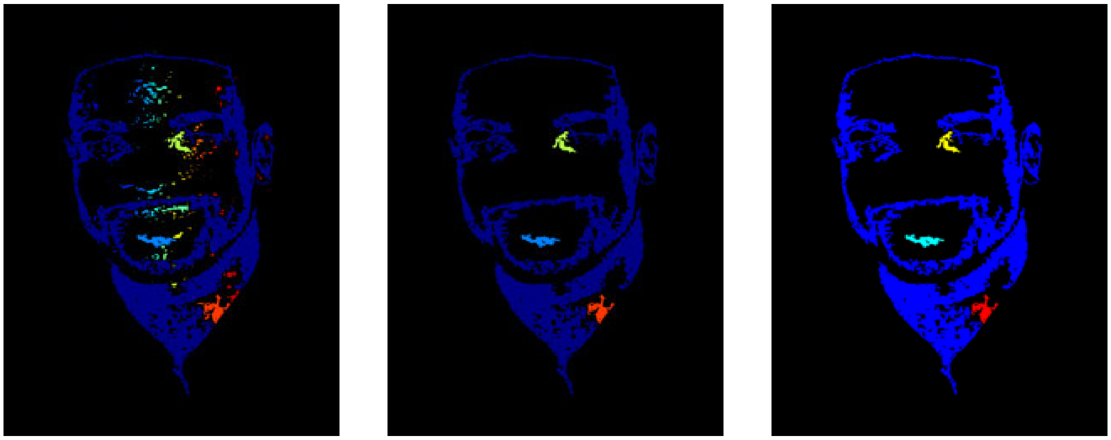
A slight problem occurs due to the exposed part of the body below the face, since it is detected as part of the face. But the real face is still successfully detected within that region. This extra detection occurs since, as it can be seen in the label images (Figures 7.3.4, 7.3.5 and 7.3.6), the facial portion and the portion below the face are "connected".

Image: black4.jpg

Figure 7.4.1: Original Image

Results:

Figures 7.4.2 & 7.4.3: Skin-likelihood Image & Binary Image



Figures 7.4.4, 7.4.5 & 7.4.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.4.7: Detected Image

Discussion:

This result is similar to the one in Figure 7.3.7; the exposed part below the face is detected to be a part of the face.

Image: chinesecouple.jpg

Figure 7.5.1: Original Image

Results:

Figures 7.5.2 & 7.5.3: Skin-likelihood Image & Binary Image

Figures 7.5.4, 7.5.5 & 7.5.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

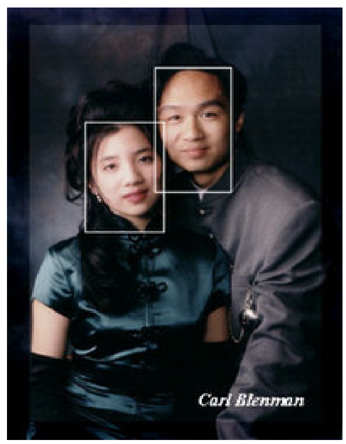


Figure 7.5.7: Detected Image

Discussion:

As can be seen in Figure 7.5.7, both the faces are detected, implying that this algorithm can be used to detect multiple faces.

Image: grp7.jpg

Figure 7.6.1: Original Image

Results:

Figures 7.6.2 & 7.6.3: Skin-likelihood Image & Binary Image

Figures 7.6.4, 7.6.5 & 7.6.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.6.7: Detected Image

Discussion:

Once again, multiple faces are detected. However, there is an “over detection”, as non-skin regions are detected to be the face.

False Positives

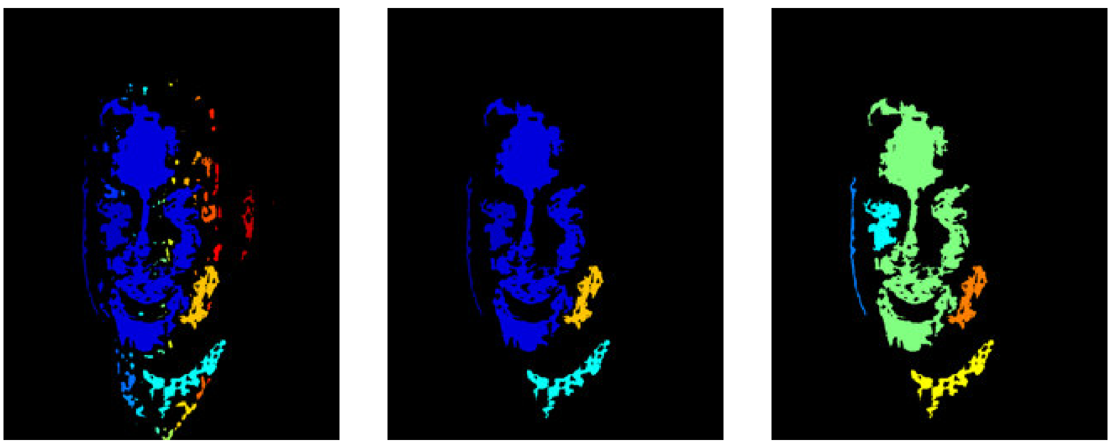
False Positive images are the detected images which contain non-face regions. Examples of non-face regions are: similar skin-colored background, other (non-face) parts of the body, and even partially detected faces, i.e. incomplete faces (like only the nose or the eye). The examples below would give a better explanation of False Positives.

Image: black2.jpg

Figure 7.7.1: Original Image

Results:

Figures 7.7.2 & 7.7.3: Skin-likelihood Image & Binary Image



Figures 7.7.4, 7.7.5 & 7.7.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

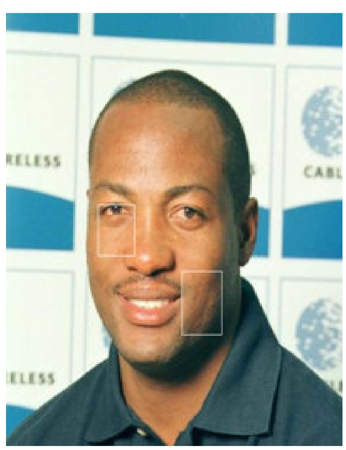


Figure 7.7.7: Detected Image

Discussion:

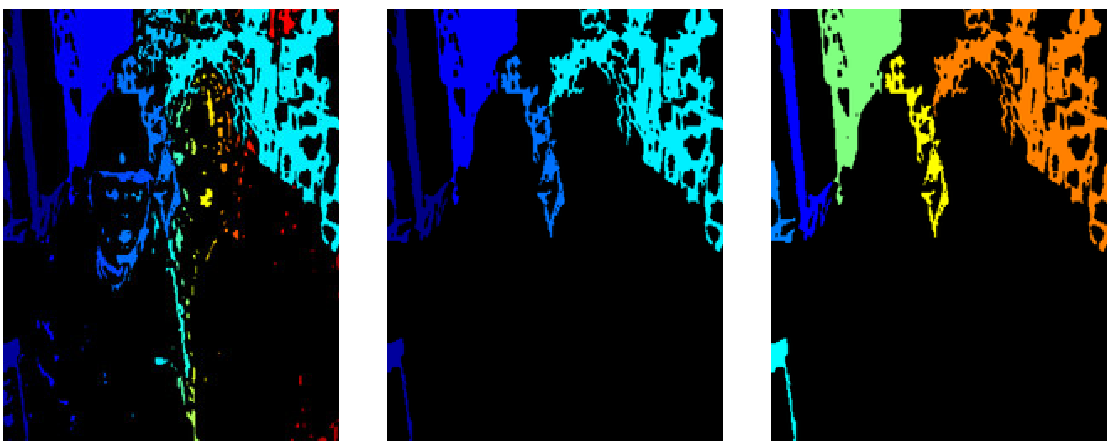
Only part of the face are detected here (Figure 7.7.7), which is of course an error. The reasons for false or erroneous detections will be discussed at the end of this chapter.

Image: grp2.jpg

Figure 7.8.1: Original Image

Results:

Figures 7.8.2 & 7.8.3: Skin-likelihood Image & Binary Image



Figures 7.8.4, 7.8.5 & 7.8.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.8.7: Detected Image

Discussion:

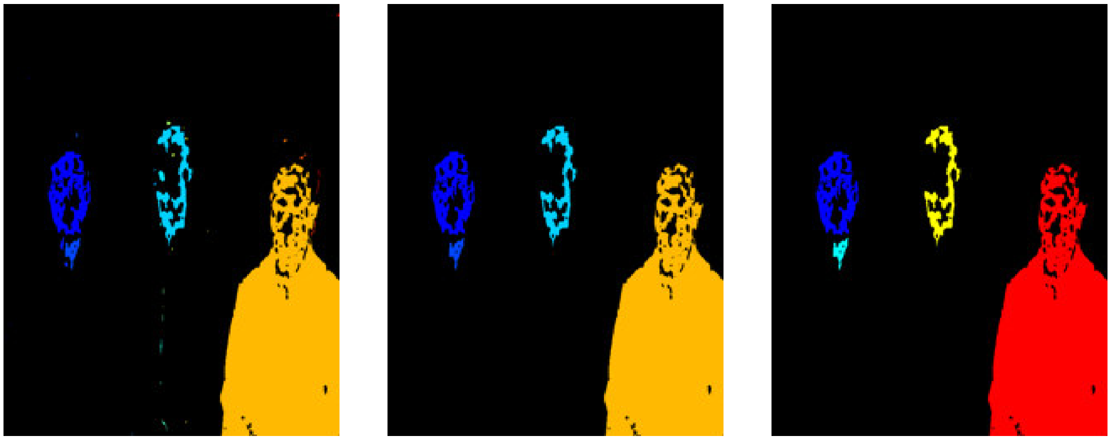
The detection here produces an even disastrous result, as virtually no facial region is detected.

Image: grp4.jpg

Figure 7.9.1: Original Image

Results:

Figures 7.9.2 & 7.9.3: Skin-likelihood Image & Binary Image



Figures 7.9.4, 7.9.5 & 7.9.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.9.7: Detected Image

Discussion:

Although three faces are present, only one is detected. But the main problem is the region below the face, the part of the neck detected as a face.

False Negatives

False Negatives are situations where a region that is *supposed to be a face* is not detected. Interestingly, an image can contain *both* False Positives and False Negatives, as can be seen in the examples under the False Positives section above. The examples shown below are more specifically targeted for False Negatives.

Image: caucasian1.jpg

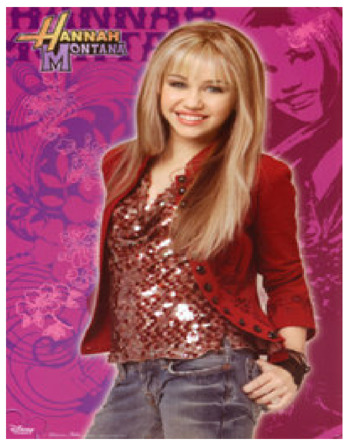
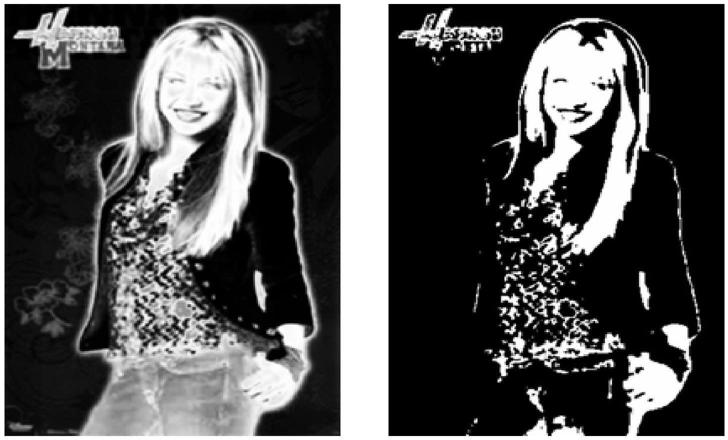


Figure 7.10.1: Original Image

Results:



Figures 7.10.2 & 7.10.3: Skin-likelihood Image & Binary Image

Figures 7.10.4, 7.10.5 & 7.10.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Discussion:

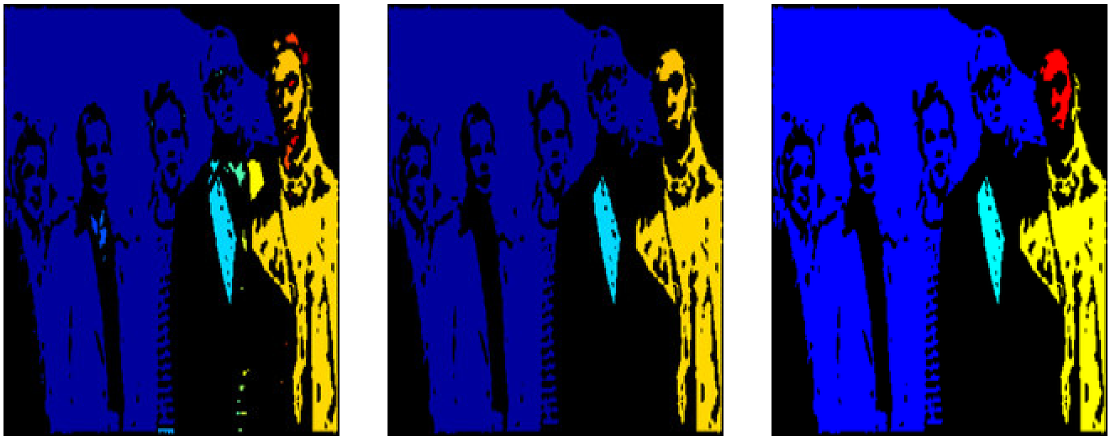
There is no “detected image” in this example. Although the labeled images separate the face, the part of the blonde hair included as a face created the negative result.

Image: grp3.jpg

Figure 7.11.11: Original Image

Results:


Figures 7.11.2 & 7.11.3: Skin-likelihood Image & Binary Image



Figures 7.11.4, 7.11.5 & 7.11.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

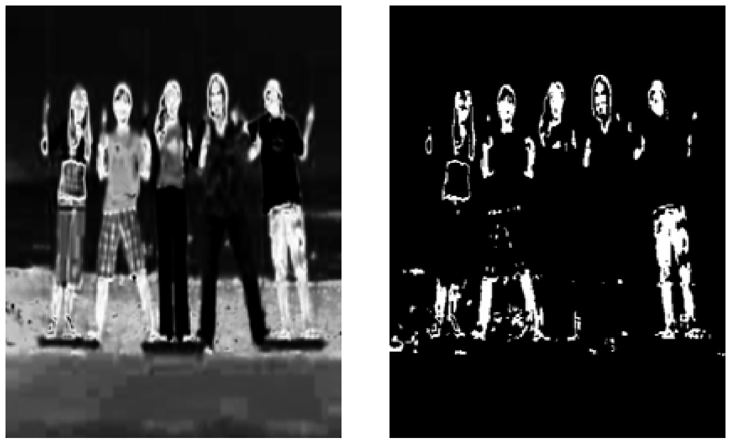
Discussion:

This example also has no “detected image”. In labeled images, four of the faces are “mixed” with the background, thus violating the cross-correlation coefficient and the region ratio. This is a very good example of the biggest problem that was encountered in this thesis, the binarization of the skin-likelihood image.

Image: grp1.jpg

Figure 7.12.1: Original Image

Results:



Figures 7.12.2 & 7.12.3: Skin-likelihood Image & Binary Image

Figures 7.12.4, 7.12.5 & 7.12.6: Labeled Image, Short-listed Labeled Image & Relabeled Image

Figure 7.12.7: (Partially) Detected Image

Discussion:

It might be a little hard to see the result in Figure 7.12.7 since the background is whitish and the detection is highlighted with a white box. The result highlights only two faces successfully out of the five. Once again, the binarization problem comes in here.

Overall Discussion

As the results show, the biggest problem was binarization. Since a global threshold value was used (thus making the system static), it is quite obvious that the binarization *will* miss out facial regions. Thus the efficiency of the program is even less than 50%. The binarization has to be made dynamic to greatly improve the results.

Another problem is with the experimented value of the red and blue means. Since only 27 skin samples were used from the three races (Asian, African and Caucasian), the result may not be robust. Increasing the number of skin samples can increase the efficiency. This is quite important because the skin-likelihood formula itself may not detect a skin to be a skin.

CHAPTER VIII

CONCLUSION

The retrieval of images containing human activities requires the ability to detect human and in *particular*, to detect *human faces* in color images in complex background. In our thesis, we implemented the method for detecting frontal human faces in color image, which first determines the skin color regions and then it locates faces within those regions by template matching such that we can detect frontal human face.

The contributions of this article are:

1. transforming a color image to a gray scale image where the gray value at a pixel represents the likelihood of the pixel belonging to the skin.
2. a thresholding method to segment a gray scale image into regions of different intensities and
3. locating face from those regions (representing the skin) by template matching.

Our proposed method of face detection has been tested by experiments. And the experimental results show that this method can detect faces in different images from different sources in an accurate and efficient manner.

We implemented the method on **32 images** and according to the results we can conclude that it is **46.25% accurate**.

We got *almost 47%* accuracy as we encountered a few *misses* in the proposed method. The misses usually included regions with a similar skin likelihood values and those regions that are definitely skin regions but they correspond to other parts of the body such as the neck, arms, and legs.

Other misses occurred as we set constraints of having one or more holes in skin regions to be considered for processing.

Therefore, we plan to further improve our method such that there will be as **less misses** as possible and hence the *accuracy* can be **increased**. In the next chapter, we discuss out further improvements in brief.

In addition, as our current method of face detection is limited to the detection of **frontal** human faces, so we are willing to do some advancement. In our future research, we wish to extend the template matching process to **side-face detection**. We discuss our advancements in the coming chapter.

CHAPTER IX

APPLICATIONS

Face detection technology can be useful and necessary in a wide range of applications, including biometric identification, video conferencing, indexing of image and video databases, and intelligent human–computer interfaces. In this section we give a brief presentation of some of these applications.

As mentioned earlier, face detection is most widely used as a preprocessor in face recognition systems. Face recognition is one of many possible approaches to biometric identification; thus many biometric systems are based on face recognition in combination with other biometric features such as voice or fingerprints. In BioID [39], a model-based face detection method based on edge information [40] is used as a preprocessor in a biometric systems which combines face, voice, and lip movement recognition. Other biometric systems using face detection include template-based methods in the CSIRO PC-Check system [41] and Eigenface methods [42,43] in FaceID from Viisage Technologies.

With the increasing amount of digital images available on the Internet and the use of digital video in databases, face detection has become an important part of many content based image retrieval (CBIR) systems. The neural network-based face detection system of Rowley *et al*. [44] is used as a part of an image search ngine for the World Wide Web in WebSeer [45]. The idea behind CBIR systems using face detection is that faces represent an important cue for image content; thus digital video libraries consisting of terabytes of video and audio information have also perceived the importance of face detection technology.

One such example is the Informedia project [46] which provides search and retrieval of TV news and documentary broadcasts. Name-It [47] also processes news videos, but is focused on associating names with faces. Both systems use the face detection system of
Rowley *et al*. [48].

In video conferencing systems, there is a need to automatically control the camera in such a way that the current speaker always has the focus. One simple approach to this is to guide the camera based on sound or simple cues such as motion and skin color. A more complex approach is taken by Wang *et al*. [49], who propose an automatic video conferencing system consisting of multiple cameras, where decisions involving which camera to use are based on an estimate of the head's gazing angle. Human faces are detected using features derived from motion, contour geometry, color, and facial analysis.

The gazing angle is computed based on the hairline (the border between an individual's hair and skin). Approaches which focus on single camera control include LISTEN [50], SAVI [51], LAFTER [52], Rits Eye [53], and the system of Morimoto and Flickner [54].

CHAPTER X

FUTURE WORK

We plan to further improve our method. We intend to make the binarization process more efficient. This means that we will do binarization using *adaptive threshloding*. Currently we are using *global thresholding* and we use an experimented value for the process. But in future, we plan to do adaptive thresholding. Using this technique of adaptive thresholding, many images yield good results; the skin-colored regions are effectively segmented from the non-skin colored regions. Therefore the chance of missing a face will be reduced. In **Figure 10.1** the face of the person has not been detected and it is missed. But if we use adaptive thresholding process then this error can be fixed and the persons face will be detected thus reducing missed faces and thus increasing accuracy.

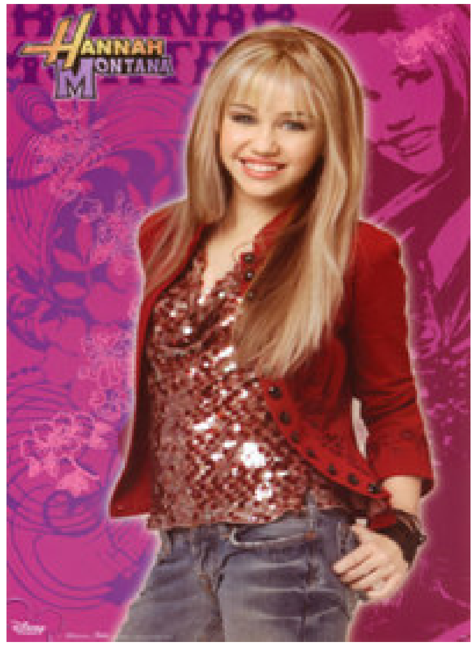


Figure 10.1 Face not detected

We will also improve candidate surface by *clipping* off regions below the face e.g. neck. If we look at **Figure 10.2** then we will notice that the neck of the person has been considered as part of the face. And therefore the neck is also included in the rectangle. But the neck is not a face region and this is an error that we need to fix. We plan to fix the error by clipping off any regions that are not a face region but mistakenly being considered as a face region. And we will do that by clipping to further improve our detection.

Figure 10.2 The neck of the person considered part of the face.

As we will improve our method and increase its efficiency, our next job will be to do a couple of advancements.

Our advancement will be:

- **Side Face Detection**
- **Face Recognition**

We will encounter many images with the side face of individuals. Therefore as an advancement of our thesis, we will propose a technique such that *side face detection* is also possible. Currently we are concentrating on detecting frontal human face but out extension will be side face detection as well.

In addition, we will also concentrate on *face recognition.* Face detection is an input to face recognition - unless a face is detected, it cannot be recognized. Therefore, a good advancement will be face recognition. Below we have given a brief idea of face recognition.

10.1 Facial Recognition System

A facial recognition system is a computer driven application that automatically identifies or verifies a person from a digital still or video image. In order to do that, selected facial features in the live image is compared with a facial database [55].

Facial recognition is typically used for security systems and it can be compared to other biometrics such as finger print or eye iris recognition system. Eigenface, fisherface, the Hidden Markov model and the neuronal motivated dynamic link matching are among the popular recognition algorithms. With the newly emerging trend of the three dimensional face recognition it has been possible to achieve previously unseen accuracies. One another emerging trend uses the visual details of the skin as it captured in standard digital or scanned images.

The facial recognition is however being used by many notable users:

- The London Borough of Newman, in the UK, has a facial recognition system that is built into their borough-wide CCTV system.

- The German Federal Police uses a facial recognition system to allow voluntary subscribers to pass fully automated border controls at Frankfurt Rhein- Main International Airport. It is necessary that the subscribers are European Union or Swiss citizens.
- Griffin investigations are famous for its recognition system that is used by casinos to catch card counters and other blacklisted individuals.

- The Australian customs service has an automated border processing system called Smart Gate that uses facial recognition. The system compares the face of the individual with the image in the e-passport microchip, certifying that the holder of the passport is the rightful owner.

In addition to being used for security system, a number of other applications have also been found out for facial recognition system.

- At Super Bowl XXXV in January 2001, police in Tampa Bay, Florida used Fault to search for potential criminals and terrorists in attendance at the event. (It found that 19 people were pending with arrest warrants).

- In the 2000 President Election, the Mexican Government employed facial recognition software in order to prevent voter fraud. This was done as there were some individuals who were registering to vote under several different names in an attempt to place multiple votes. Thus, facial recognition system was used for that purpose and by comparing new facial images to those images that were already saved in the voter database; authorities were able to reduce duplicate registrations.
- Similar technologies are being used in the United States to prevent people from obtaining fake identification cards and driver's license.

There are a number of potential uses of facial recognition system that are currently being developed. For instance, this technology can be used as a security measure at ATM's; it can be in such a way- instead of using a bank card or a personal identification number (PIN), the ATM would capture an image of a face and then compare it with the photo in the bank database to confirm identity. Similar concept can be applied to computers, the face i.e. the digital image of an individual can be captured by using a web cam and then using this technology the individual can easily log-in without using password.

However, there are a few criticisms to this system.

- Camera technology designed to spot potential terrorists by their facial terrorists at airport failed its first major test at Boston's Logan airport.

- Critics of this technology complain that the London Borough of Newham Scheme has as of 2004, never recognized a single criminal, despite several criminals in the systems database living in the Borough and the system having been running for several years. "Not once, as far as the police know, has Newham's automatic facial recognition system spotted a live target!"

In addition, despite the potential benefits of this technology, many citizens are aware about the fact that their privacy will be invaded. Some even fear that this might lead to a "total surveillance society", with the Government and other authorities having the ability to know where a person is and what that person is doing, at all times.

But whatever criticisms may arise from such a technology, the benefits that we get from such a technology are immense and among all the different biometric technology, facial recognition system has a great advantage of not requiring aid from the test subject. Properly designed systems installed in airports, multiplexers and other public places can detect the presence of criminals among the crowd. Other biometrics that includes fingerprint, iris recognition and speech recognition cannot perform this kind of mass scanning. Therefore, facial recognition system has been very effective.

BIBILOGRAPHY

[1] http://en.wikipedia.org/wiki/Face_detection

[2] http://www.facedetection.com/facedetection/techniques.htm

[3] T. Sakai, M. Nagao, and T. Kanade, Computer analysis and classification of photographs of human faces, in Proc. First USA—Japan Computer Conference, 1972, p. 2.7.

[4] 14. R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: A survey, Proc. IEEE 83, 5, 1995.

[5] R. Brunelli and T. Poggio, Face recognition: Feature versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15, 1993, 1042–1052.

[6] D. Valentin, H. Abdi, A. J. O'Toole, and G. Cottrell, Connectionist models of face processing: A survey, Pattern Recog. 27, 1994, 1209–1230.

[7] H. Demirel, T. J. Clarke, and P. J. K. Cheung, Adaptive automatic facial feature segmentation, in IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gestur Recognition, Vermont, Oct. 1996, pp. 277–282.

[8] D. Valentin, H. Abdi, A. J. O'Toole, and G. Cottrell, Connectionist models of face processing: A survey, Pattern Recog. 27, 1994, 1209–1230.

[9] Junhong Liu, Face Detection - Advanced Topics in Information Processing

[10] http://en.wikipedia.org/wiki/Binary_image

[11] http://en.wikipedia.org/wiki/Grayscale

[12] http://en.wikipedia.org/wiki/Pixel

[13] http://en.wikipedia.org/wiki/RGB_color_model

[14] http://en.wikipedia.org/wiki/Luminance

[15] http://en.wikipedia.org/wiki/Segmentation

[16] http://www.itl.nist.gov/div898/handbook/pmc/section5/pmc541.htm

[17] http://www.cee.hw.ac.uk/hipr/html/threshld.html

[18] http://www.cee.hw.ac.uk/hipr/html/adpthrsh.html

[19] http://en.wikipedia.org/wiki/Connected_component_labeling

[20] R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: A survey, Proc. IEEE 83, 5, 1995.

[21] A Samal and P. A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: a survey, Pattern Recog. 25, 1992, 65–77.

[22] R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: A survey, Proc. IEEE 83, 5, 1995.

[23] M-H. Yang, D.Kriegman, and Ahuja, "Detecting faces in Images: A Survey", IEEE Trans. Patern Analysis and Machine Intelligence, vol.24, no.1, Jan.2002.

[24] E. Hjelm and B.K.Low, "Face Detection: A Survey", Computer Vision and Image Understanding, vol.83,110.3, pp.236-274, Sept.2001

[25] H. A. Rowley, S. Baluja, and Kanade, "Neural network-Based Face Detection", IEEE Trans. Patern Analysis and Machine Intelligence, vol.20, pp.23-28, Jan.1998.

[26] H. A. Rowley, S. Baluja, and Kanade, "Rotation Invariant Neural network-Based Face Detection", Proc IEEE Conf. Computer Vision and Patern Recognition, pp.38-44, Jan.1998.

[27] K.K. Sung and T. Poggio, "Example-Based Learning and View-Based Human Face Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol20, pp.39-51, Jan.1998.

[28] H. Schnelderman and T Kanade, " A statistical Method for 3D object Detection Applied to Faces and Cars", IEEE Conf.Computer Vision and Patern Recognition, pp.746-751. 2000.

[29] K. C. Yow and R. Copolla, "Feature-based Human Face Detection", Image and Vision Computing, vol 15, no.9, pp.713-735, Sept.2000.

[30] D. Maio and D.Maltoni, "Real-time Face Location on Gray-scale Static Images", Pattern Recognition, vol.33, no.9, pp.1525-1539, Sept.2000.

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

[32] D.Maio and D.Maltoni, "Real-time Face Location on Gray-scale Static Images", Pattern Recognition, vol.33, no.9, pp.1525-1539, Sept.2000.

[33] **Henry Chang and Ulises Robles, "Face Detection"**

http:// www-cs-students.stanford.edu/~robles/ee368/main.html

[34] Jie Yang and Alex Waibel, "A Real-Time Face Tracker", CMU CS Technical Report.

[35] J. Cai & A. Goshtasby & C. Yu, "Detecting Human Faces in Color Images", Wright State University, U. of Illinois.

[36] G. Wyszecki and W.S. Styles. Color Science: Concepts and Methods, Quantitative Data and Formulae, second edition, John Wiley & Sons, New York 1982.

[37] Y. Gong and M. Sakauchi, "Detection of regions matching specified chromatic features", Computer Vision and Image Understanding, vol. 61, no. 2, 1995, pp 263 - 269

[38] R. Ramesh, Kasturi R. and Schunck B., Machine Vision, pp 31 - 51, McGraw Hill, New York 1995.

[39] R. W. Frischolz and U. Dieckmann, BioID: A multimodal biometric identification system, IEEE Comput. 33(2), 2000.

[40] G. A. Klanderman, D. P. Huttenlocher, andW. J. Rucklidge, Comparing images using the Hausdorff distance, IEEE Trans. Pattern Anal. Mach. Intell. 1993, 850–863.

[41] G. E. Poulton, N. A. Oakes, D. G. Geers, and R.-Y. Qiao, The CSIRO PC-check system, in Proceedings Second International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA), 1999.

[42] A Pentland, B. Moghaddam, and T. Strarner, View-based and modular eigenspaces for face recognition, in IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 1994.

[43] M. Turk and A. Pentland, Eigenfaces for recognition, J. Cog. Neurosci. 3, 1991, 71–86.

[44] H. A. Rowley, S. Baluja, and T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20, January 1998, 23–38.

[45] C. Frankel, M. J. Swain, and V. Athitsos, WebSeer: An Image Search Engine for the World Wide Web, Technical Report TR 96-14, Computer Science Department, Univ. of Chicago, 1996.

[46] H. D. Wactlar, T. Kanade, M. A. Smith, and S. M. Stevens, Intelligent access to digital video: Informedia project, IEEE Comput. 29(5), 1996, 46–52.

[47] S. Satoh, Y. Nakamura, and T. Kanade, Name-It: Naming and detecting faces in news videos, IEEE Multimedia 6, 1999, 22–35.

[48] H. A. Rowley, S. Baluja, and T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20, January 1998, 23–38.

[49] C.Wang, S. M. Griebel, and M. S. Brandstein, Robust automatic video-conferencing with multiple cameras and microphones, in Proc. IEEE International Conference on Multimedia and Expo, 2000.

[50] M. Collobert, R. Feraud, G. Le Tourneur, O. Bernier, J. E. Viallet, Y. Mahieux, and D. Collobert, LISTEN: A system for locating and tracking individual speaker, in IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition, Vermont, 1996.

[51] R. Herpers, G. Verghese, K. Derpanis, R. McCready, J. MacLean, A. Jepson, and J. K. Tsotsos, Detection and tracking of faces in real environments, in Proc. IEEE International Workshop on Recognition,

Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999.

[52] N. Oliver, A. Pentland, and F. B´erard, LAFTER: A real-time face and lips tracker with facial expression recognition, Pattern Recog. 33, 2000, 1369–1382.

[53] G. Xu and T. Sugimoto, Rits eye: A software-based system for realtime face detection and tracking using pan-tilt-zoom controllable camera, in Proc. of International Conference on Pattern Recognition, 1998.

[54] C. Morimoto and M. Flickner, Real-time multiple face detection using active illumination, in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[55] http://en.wikipedia.org/wiki/Facial_recognition_system