ABSTRACT

GUNDUZ, AYSEGUL. Compression and Transmission of Facial Images Over Very Narrowband Wireless Channels. (Under the direction of Dr. Hamid Krim.)

Law enforcement officers on mobile duty are often confronted with ID authentication of subjects, requiring the transmission of a driver's license picture over wireless channels with very narrow bandwidths. To access mug shots in a reliable and timely manner, real time compression and decompression methods with high compression ratios are required at the server database and at the mobile client unit. This thesis presents a methodology, which minimizes the size of the data sent over the channel by locally storing common features of the human face in the client computers. Pre-processing of server database images, such as facial feature extraction, are used to extract these common facial features, and are obtained via topological methods, in particular, via ravine extraction. The implemented file transfer protocols are based on basic TCP/IP client-server models and make use of socket programming. Experimental results show a 4x improvement in transfer time over typically saturated channels.

COMPRESSION AND TRANSMISSION OF FACIAL IMAGES OVER VERY NARROWBAND WIRELESS CHANNELS

by
AYSEGUL GUNDUZ

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

ELECTRICAL AND COMPUTER ENGINEERING

Raleigh

2003

APPROVED BY:

Dr. Alexandra Duel-Hallen	Dr. Christopher G. Healey
P. Allan Sadowski	Dr. Hamid Krim Chair of Advisory Committee

To my parents

Biography

Ayşegül Gündüz was born in Ankara, Turkey, on January 21st, 1981. In 1997, she represented her country in the European Council of International Schools Mathematics Championship, in which she became the overall winner. The same year, following her graduation from T.E.D. Ankara College High School, she attended the Electrical and Electronics Engineering Department at Middle East Technical University, Ankara, Turkey.

Ayşegül graduated from Middle East Technical University with honors, and received her B.S. degree in June 2001. She started her graduate education in the Master of Science program of the Electrical and Computer Engineering Department at North Carolina State University in Fall 2001. Since November 2001, she has been conducting research on image processing in the Vision, Information, and Statistical Signal Theories and Applications Group, under the supervision of Dr. Hamid Krim.

Acknowledgements

I would like to express my appreciation and respect towards my advisor, Dr. Hamid Krim, for his guidance has shaped this work and has presented me valuable skills, which I will take along with me throughout my academic journey. This gratitude extends to the members of the advisory committee, Dr. Alexandra Duel-Hallen and Dr. Christopher Healey, for their time and attention in the evaluation of the thesis, especially to Allan Sadowski, who has given me his endless support and the opportunity to work on this project. I also owe great thanks to Rodney Spell for his invaluable help on the wireless network communication. The National Science Foundation and North Carolina State Highway Patrol are also gratefully acknowledged.My deepest gratitude goes to my parents, Fahriye and Şükrü, who have devoted their lives to providing me with the best education possible. Every single achievement in my life belongs to them and whenever they should feel proud of me, they should rather be proud of themselves.

I would also like to thank my sister Zeynep, not only for being my lifelong best friend, lighting my path with her experiences, but also for being the prettier sister and letting me be the smarter one.

Finally, I would like to thank Rizwan for always bringing out the best in me and believing in me more than I believe myself.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human face is a fascinating image, as it is the distinguishing characteristic of all humans on earth. The fact that almost all identification cards bear a photograph of a person is due to the global acceptance of a facial image as an evidence of identity. Humans are capable of accurately matching a frontal facial image and a person to affirm their identity or difference (except for prosopagnosia patients [1]). This process throughout the sequel will be referred as *person identification* or *authentication.*

In addition to matching faces, the detection and recognition of faces are some of the most remarkable abilities of the human brain. The *human recognition* of faces is the process of perceiving someone as previously known or seen, and should not to be confused with the concept of *identification* as defined above.

The human perception of faces has been studied in the fields of psychophysics and neuroscience [2, 3, 4, 5, 6] for many years. Inspired by the human perception, *facial image compression for person identification* is an important problem in computer vision and image processing in facial image analysis (although more attention has been on *automatic face recognition* [7, 8]). All human faces share many common features, each with its dedicated function. This influences the notion of some common template of all human faces, and the notion of some distinguishing features for identifiability. Two main

approaches have been taken in facial image compression: *low-dimensional representation of faces* and *feature-based compression*.

The first technique, introduced by Sirovich and Kirby [9, 10], involves the application of the *Karhunen-Loève expansion* [11] (more widely known in the engineering literature as *principal component analysis* or *eigenvector decomposition*) to a set of facial images. The goal of the approach is to represent a face in terms of an optimal coordinate system whose dimensionality is lower than the dimensionality of the face space. The set of basis vectors, which make up this coordinate system are the eigenvectors of the covariance matrix of the set of images. Each eigenvector is an actual picture, which is called an *eigenpicture.* Sirovich and Kirby used the fact that a face may be approximately reconstructed from its eigenvalues corresponding to the set of principal eigenpictures. Thus, with the a priori knowledge of the optimal coordinate system or the eigenpictures, the eigenvalues are sufficient to represent a face. The approach was adopted by Turk and Pentland [12], who renamed the eigenpictures as *eigenfaces*, for face recognition. However, in [13], Penev and Sirovich discuss the dimensionality of the optimal coordinate system necessary for an adequate representation of the identity information and the shortcomings of principal component analysis. Local feature analysis [14] is proposed as a solution.

The main idea behind feature-based approaches is that most of the identity information is contained in the facial features such as eyes, mouth, and nose. Therefore, these features should be coded in detail, and hence allocated more resources (bits), whereas, other features such as the forehead, cheeks or hair may be coded with less bits allocated. The procedure involves a high ratio lossy compression stage followed by a

refinement at the regions of facial features [15]. Vector quantization [16] and wavelet-based compression [17] have been proposed for the lossy compression stage. For these techniques, *facial feature extraction* is essential.

Various techniques have been proposed in the literature for facial feature extraction, which may be classified in four main groups: geometry-based, template-based, color segmentation-based, and appearance-based approaches. In *geometry-based approaches*, the features are extracted using geometric information such as relative positions and sizes of the face components. The first automatic facial feature extraction technique, proposed by Kanade [18], falls into this category. The features are extracted from the vertical and horizontal integral projections or edge maps of the original image. The eyes, the mouth and the nose base are localized using the vertical edge map, whereas, the nose, and the left and right boundaries of the face are localized using the horizontal edge map. Valley detection filters [19], and application of Hough transform [20] are also classified within this group. These techniques require thresholding, which, given the prevailing sensitivity, may adversely affect the achieved performance.

Template-based approaches match facial components to previously designed templates using appropriate energy functionals. The best match of a template in the facial image will yield the minimum energy. Deformable templates for facial feature extraction were first proposed by Yuille et al. [21]. Active contours [22], and active meshes [23] were subsequently used for purposes of feature extraction. These algorithms require a priori template modeling, in addition to their computational costs, which clearly affect their performance. Genetic algorithms have been proposed for more efficient searching times in template matching [24, 25].

Color segmentation techniques [26], on the other hand, make use of skin color to isolate the face. Any non-skin color region within the face is viewed as a candidate for eyes and/or mouth. The projections of images into different color spaces have also been analyzed, and in [27] it is shown that some features can easily be extracted in specific color system coordinates. The performance of such techniques on facial image databases is rather limited, due to the diversity of ethnical backgrounds.

Finally, *appearance-based approaches* aim to find basis vectors to represent a face. The concept of "feature" in these approaches differs from simple facial features such as eyes and mouth. Any extracted characteristic from the image is referred to a feature. Methods such as principal component analysis (as mentioned earlier), independent component analysis, and Gabor-wavelets [28], [29] are used to extract the feature vector. These approaches are commonly used for face recognition rather than person identification.

1.1 Objective

Law enforcement officers on mobile duty are often confronted with ID authentication of subjects, requiring the transmission of a driver's license picture over wireless channels with very narrow bandwidths. A mug shot picture from databases (e.g. Dept. of Motor Vehicles) in hand provides officers with person identification in cases of absence of an ID and/or authentication in case of a suspicion. Wireless transmission is the solution for communication with mobile computers. The wireless network channels of law enforcements, however, have to be shared among various agencies and the channels run almost always at full capacity.

To access mug shots in a reliable and timely manner, a real time compression and decompression method with high compression ratios is required at the server database and at the mobile client unit. Standard compression algorithms (e.g. JPEG 2000) [30] exhibit an unacceptable performance in so far as the required transmission time is concerned (upload of 4 to 5 min). Thus, a highly specialized technique optimized around mug shot wireless transmission has to be developed. The objective of this work is to propose a robust, efficient, and simple image compression technique optimized for ID pictures and adapted to a communication protocol used by the North Carolina State Highway Patrol (NCSHP) over a very narrowband channel (<10 kbps).

Human faces share a common template, which is composed of two eyes, one nose, one mouth, and so forth. Moreover, their locations relative to each other on the average are known. Only shape, size and distance between these features vary from person to person. The proposed methodology is based on using common features of face images, i.e., finding a template at some level that is common to all faces, which will be available at all client computers. Since the compression technique is feature-based, a fast and reliable feature extraction method is essential. For this purpose, a methodology that makes use of the topological features of the human face in order to localize the eyes and mouth on a driver's license type of image is proposed.

1.2 Organization

The next chapter is dedicated to the extraction methodology of facial features. Some fundamental background is covered, with relations and interpretations for the facial feature extraction process.

Chapter 3 introduces the facial image compression approach proposed in this work. Particular attention is given to level set decomposition of facial images and their reconstruction from the principal level sets. Compression techniques for coding the principal level sets are also presented.

Chapter 4 describes the overall transmission of facial images over the NCSHP wireless communication channels. The general functioning is presented and illustrated, with the wireless communication network schematic and the end user programs.

Finally, Chapter 5 focuses on the experimental results of our compression technique, along with their discussion.

Chapter 2

Facial Feature Extraction

Human facial features play a significant role in perceiving faces. Neurophysiological research and studies have determined that eyes, mouth, and nose are amongst the most important features for recognition [1]. Thus, when a human face is represented as an image, it is very natural for these features to depict distinguishing characteristics not present in other facial components such as forehead, cheeks and chin. The eyes, the mouth, and the nostrils are the local minima of a facial image, whereas, the tip of the nose is a local maximum. However, for the purpose of compression herein, the facial features of interest are only the eyes and the mouth.

When a facial intensity image is represented as a surface, the brightness values form deep valleys in the facial feature regions, allowing topological methods to be used for their localization. The key idea in the methodology is to model the eyes and mouth as ravines on the image surface. For this purpose, we first present a brief background on differential geometry and topology basics in this chapter, which is followed by the feature extraction technique we developed.

2.1 Differential Geometry And Topology Fundamentals

In differential geometry, the most characterizing feature of a curve is its *curvature*. Curvature is a measure of rate of change of the direction of the tangent vector, or simply

a measure of the bending of the curve. The magnitude of curvature at a point is inversely proportional to the radius of the circle that fits the curve at that point [31].

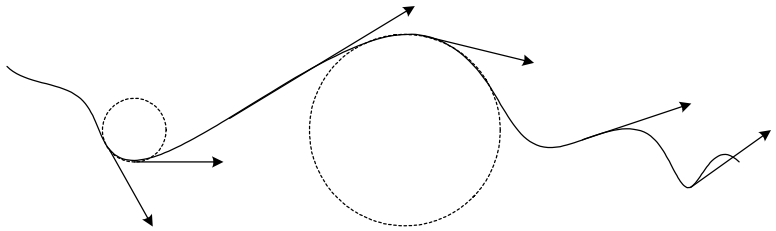


Figure 2.1: The curvature tells us which way the tangent vector is turning and how fast. Its magnitude is the reciprocal of the osculating circles.

The bending of regular surfaces in R^3, on the other hand, is measured via a linear operator called the *shape operator*, which estimates the change of surface normal from point to point [32]. For a regular surface S, F_1 and F_2 represent *the first and second fundamental forms*, respectively, given by:

$$F_1 = \begin{bmatrix} 1+S_x^2 & S_x S_y \\ S_x S_y & 1+S_y^2 \end{bmatrix}, \tag{1}$$

$$F_2 = \frac{1}{\sqrt{1+S_x{}^2+S_y{}^2}} \begin{bmatrix} S_{xx} & S_{xy} \\ S_{xy} & S_{yy} \end{bmatrix}, \tag{2}$$

where subscripts represent partial derivatives with respect to the variable (x or y). The first fundamental form determines the arclength of a curve on a surface, whereas, the second measures how far the surface is from being a plane.

$$W = F_1^{-1} \cdot F_2 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}, \tag{3}$$

is the matrix of the shape operator, and equation (3) is known as the Weingarten equation. [32, 33].

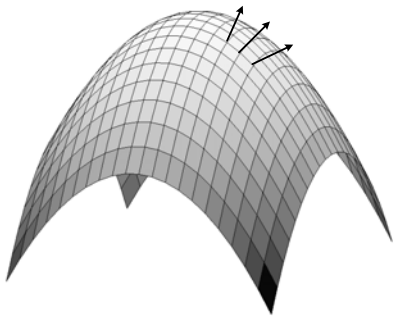


Figure 2.2: The shape operator measures the change of surface normal from point to point on the surface.

Consider a point P on a regular surface S. There are an infinite number of planes passing through the normal vector at P and intersecting S at plane curves, i.e., whose intersections with S form plane curves. The curvatures of these plane curves are called the *normal curvatures* of S at P. The signs of normal curvatures depend on the surface normal vector. The *principal curvatures*, k_{max} and k_{min} are the most positive (upward) and the most negative (downward) of the normal curvatures, respectively. The maximum and minimum curvatures occur at orthogonal directions called the *principal directions*, $\vec{d}_{\max}$ and $\vec{d}_{\min}$. The principal curvatures are the eigenvalues of the shape operator and the principal directions are the corresponding eigenvectors [31].

Other important curvatures of a surface are the *Gaussian* and *mean curvatures*, which may be computed, respectively, as follows:

$$K = \det(W), \tag{4}$$

$$H = \frac{1}{2} Tr(W). \tag{5}$$

The eigenvalues and eigenvectors can be subsequently expressed as:

$$\begin{aligned} k_{\max} &= H + \sqrt{H^2 - K}, \\ k_{\min} &= H - \sqrt{H^2 - K}, \end{aligned} \tag{6}$$

$$\vec{d}_{\max} = (w_{12}, k_{\max} - w_{11}),$$
$$\vec{d}_{\min} = (w_{12}, k_{\min} - w_{11}). \tag{7}$$

A point on a surface at which the principal curvatures are equal and non-zero, is called an *umbilic point.* (In the case of zero curvatures, the point is called a *flat* point.)

Ravines are defined as non-umbilic points at which k_{max} attains a local maximum along $\vec{d}_{\max}$ [34]. A non-umbilic point *P* on the surface is, thus, a ravine point if and only if it satisfies the following conditions:

$$k_{\max}(P) > 0,$$
$$\nabla k_{\max}(P).\vec{d}_{\max} = 0, \tag{8}$$
$$\vec{d}^{\,T}_{\max}.\nabla^2 k_{\max}(P).\vec{d}_{\max} < 0.$$

A non-umbilic point *P* on a surface is called a *ridge* point if k_{min} attains a local minimum along $\vec{d}_{\min}$, i.e., satisfies:

$$k_{\min}(P) < 0,$$
$$\nabla k_{\min}(P).\vec{d}_{\min} = 0, \tag{9}$$
$$\vec{d}^{\,T}_{\min}.\nabla^2 k_{\min}(P).\vec{d}_{\min} > 0.$$

Ridges and ravines may also be defined according to the directions of the normal vector [34, 35]. Throughout, the curvatures of convex up curves are taken to be positive. Therefore, ravines have a tendency to detect the valleys on a surface, and ridges extract crests.

2.2 Feature Extraction Methodology

A two dimensional image ***I****(x,y)* may be viewed as a surface, *S* = { *(x,y,z)* : *z* = ***I****(x,y)* } , where the values of the *z*-axis is represented by the image intensity at a pixel *(x,y)*. Figure 2.3.b depicts the facial image shown in Figure 2.3.a as a surface. The features of interest, the eyes and the mouth, are composed of both crests and valleys. The valleys, however,

represent the most defining characteristics - the borders of eyes, irises and the opening of the mouth, for instance. Our goal is therefore to extract the ravine points.

(a) (b) (c)

Figure 2.3: (a) A driver's license photograph, (b) the image plotted as a surface, (c) the image with extracted background.

2.2.1 Preprocessing of Images: Background and Noise Removal

North Carolina State driver's license images have a standard blue background. Thus, the variation of background color from image to image is due to varying illumination. In order to detect the blue color information, it is essential to work in a color space that isolates brightness and color. The usage of the HSV (hue, saturation, value) space to extract color information has been proposed in [36, 37].

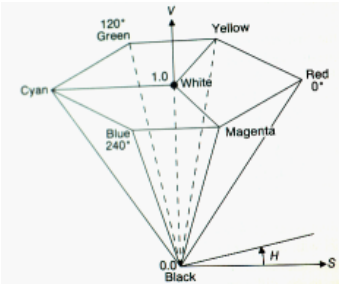


Fig 2.4: The HSV color space in a hexacone.

The HSV color space is often used by people who are selecting colors (e.g., of paints or inks) from a color wheel or palette, as it is better adapted than the RGB (red, green,

blue) color space to human color perception. As hue varies from 0 to 1.0, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red, so that there are actually red values at both 0 and 1.0. As saturation varies from 0 to 1.0, the corresponding colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness, varies from 0 to 1.0, the corresponding colors become increasingly brighter [38]. The HSV space is shown in Fig 2.3.c.

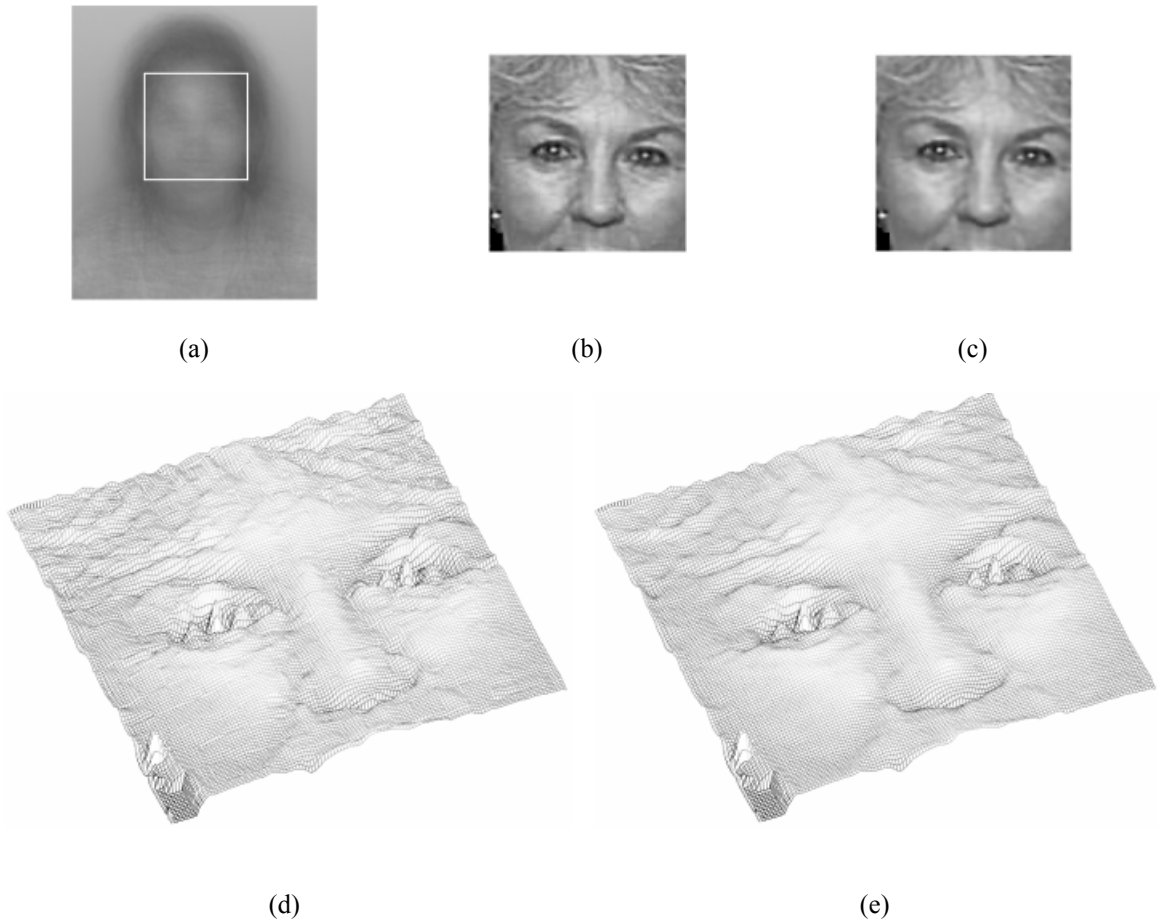
From a large set of facial images from NCSHP database, the hue value range for the background was found to be (0.35,0.68). The image in Fig 2.3.a. with removed background is shown in Fig 2.3.c. Although, false detection is possible on the boundary of the hair region or clothes, the face region is never falsely detected.

The calculation of the principal curvatures involves second derivatives, which in turn require some smoothness in the images of interest. While smoothing the image for noise removal, we would also like to preserve the facial features. Adaptive Wiener filters, which are commonly used in image restoration [39], have also been proposed for facial images. As Wiener filters are comparatively slow, the features of interest are smoothed slightly, while the effect of noise is reduced. Figure 2.5.c is the output of a two-dimensional Wiener filter applied to a box extracted from the image.

2.2.1 Extraction of the Eyes

By examining a large set of driver's license database images, which are head-shoulder images, we have determined that a box may be extracted automatically from the image with little regard to symmetry, guaranteeing the inclusion of the eyes. This is due to the face positioning on a driver's license photograph as may be observed in an average image

(of 100 Caucasian facial images) in Figure 2.5. In the white frame, ghostly blurred eyes can be observed.



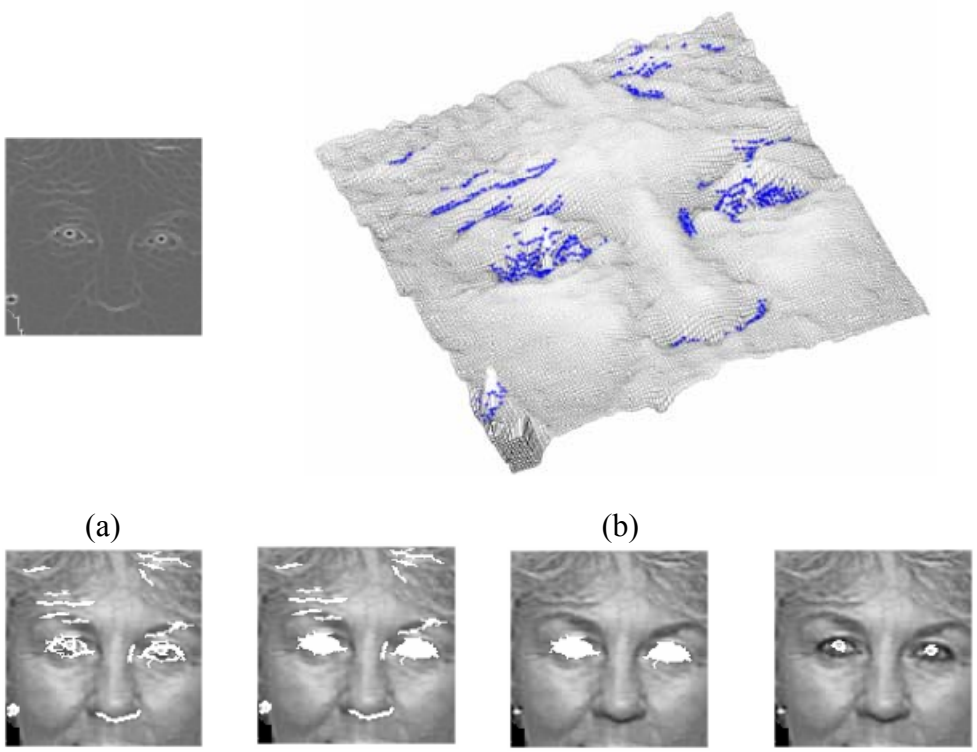
(a) (b) (c)

(d) (e)

Figure 2.5: (a) The average image of 100 driver's license images; the box to be extracted automatically contains the eye regions, (b) the box extracted from the photograph in Fig 2.3.a, (c) The smoothed surface via Wiener filtering, (c) the surface plot of "(b)", (d) the surface plot of "(c)".

Upon filtering the image, we apply the set of conditions (8) to every pixel. By restricting k_{max} to be greater than a positive threshold, we can extract the sharp bendings and hence the most significant ravines. The threshold is determined by the mean and standard deviation of k_{max}. k_{max} is depicted in Figure 2.6.a.The term $\nabla k_{\max}.\vec{d}_{\max}$ in (8) is defined as *the directional derivative of the maximum curvature (DMC)* [40]. The transition points (i.e. positive to negative and vice versa), along with the zeros constitute

the zero crossings of the DMC and thus the solution. The result of the applied algorithm is shown in Figure 2.6.b-c.



(a) (b)

(c) (d) (e) (f)

Figure 2.6: (a) The maximum curvature of each pixel in the image, (b) the extracted ravines from the surface, (c) the superposition of the ravines on the image, (d) morphological closing and filling of the extracted eyes, (e) the extracted eye regions, (f) the ravines that yield the irises.

Additional processing around each feature (e.g. eye regions) is required to provide continuity of the feature trace and fill in any gaps in borders to facilitate the localization of the features. This may be carried out in a number of ways, and the simplest is application of morphological closure and filling on the image bearing the ravines (see Figure 2.6.d). The resultant closed regions are thresholded according to their area and their normalized intensity, and the image is divided into two half planes. For every region in the left half plane, a corresponding pair is searched in the right half plane with the following constraints:

- Their horizontal positioning should be in a specified range with respect to each other,
- The difference between their lengths and widths should be smaller than a threshold,
- The distance between their vertical positioning should neither be significantly greater, nor smaller than their horizontal sizes (as the distance between two eyes is approximately the size of one eye in human faces),
- The normalized intensity of the regions should be considerably close in value.

In case of more than one eye pair candidate, the horizontal positioning of the pairs with respect to each other and the box is used for extraction. For example, if fair eyebrows are detected as an eye pair candidate, the horizontal positioning of eye and eyebrow pairs are fairly close. In this case, the algorithm picks the lower positioned pair. If the distance between two candidates is far, the positioning with respect to the box is taken as basis. The eye pair localization is then followed by computing the centroids of the extracted regions and labeling them as eye centers.

Further increasing the threshold of k_{max}, results in the extraction of irises, whose centroids can be more accurate measures for the eye centers, in cases of nonsymmetrical or insignificant eye borders. Figure 2.6.f shows how a greater threshold for the maximum curvature yields the iris boundaries.

If the alignment between two eyes deviates from the horizontal axis by more than 5^o, the image is rotated to retain horizontal alignment. An example is given in Figure 2.7.

(a) (b)

Figure 2.7: (a) A slightly tilted face in a driver's license image, (b) output of image rotation for horizontal alignment of the eyes.

2.2.3 Extraction of the Mouth

The eye locations are in turn utilized in extracting a box containing the mouth (see Figure 2.8.a). The width of the box is kept small in order to minimize the inclusion of the jaw line, which is also composed of ravines. The extracted ravines once again undergo morphological operations (see Figure 2.8.b). The region with the greatest width is detected to be the mouth. The mouth location is used together with the eye pair locations to construct the standard face triangle whose vertices are the eye and mouth centers, as shown in Figure 2.8.c. Finally, the images are resized so that the sizes of the triangles in each image are fixed to the same size.

(a) (b) (c)

Figure 2.8: (a) The mouth box extracted relative to the localization of the eyes, (b) the extracted ravines, (c) the triangle connecting the centers of the extracted eye and mouth regions.

Chapter 3

Facial Image Compression

Image compression addresses the problem of reducing the amount of data to represent an image. Yet, in the case of transmission, the goal is to reduce the amount of data to be transmitted. If a common portion of the data of all images were to be permanently available at the destination, then only the differentiating data among pictures need be transmitted. That is precisely the basis of our approach [41].

The idea behind facial feature extraction, i.e., extracting the triangle whose vertices correspond to localized eye and mouth features, and resizing every image to attain a fixed triangle size, is to obtain a geometrical template for human faces. Also, using the localization of the triangle, we can extract the facial region (of size 160x160 pixels), and in turn reduce the sizes of the images by eliminating redundant information not critical to face identification (see Figure 3.2). These facial images are further downsampled by a factor of two, and hence the resultant size of the images to be compressed is 80x80x8 bits (6.25 Kbytes).

Faces are, however, composed of other features in addition to eyes, mouth, nose etc. An undeniably important component is the human skin. In an ethnic group, the tone of skin shows more homogeneity than in a mixed population. Moreover, some auxiliary facial features are not always present in every human face, e.g., eyeglasses, mustaches and beards. Hence, in the database of driver's license images there are some groups that

exhibit more similarity than others, implying the necessity of dividing the database into subclasses. Examining a large set of database images, we determined that the major subclasses should be: Caucasians vs. African-Americans, presence vs. absence of eyeglasses, presence vs. absence of mustache or beard.

3.1 Classification of Images

Since the existing driver's license database is not to be modified, the classification of an image needs to be autonomous. The classification of Caucasian vs. African-American is achieved via thresholding the mean of the brightness of the extracted face images. A verification stage via color segmentation is feasible, but was determined to be unnecessary.

The detection of eyeglasses and mustaches is accomplished during the eye and mouth extraction stages respectively. The eyeglasses and mustaches form deep valleys on the image surface, which are again detected via ravine extraction, and their presence results in bigger areas after the morphological closure and filling stage than the cases of their absence. We defer the details on the extraction of auxiliary features to the appendix.

Finally, the mean images of all subclasses are extracted. Figure 3.1 shows the mean image of 100 Caucasians with no auxiliary features.

Figure 3.1: The mean image of 100 database photographs of Caucasians.

3.2 Image Representation via Levels Sets

Level sets are the "binary shadows" of an image and can be defined by

$$L_i^{(I)}(x,y) = \begin{cases} 1, \textit{ if } I(x,y) \geq \lambda_i \\ 0, \textit{ otherwise} \end{cases} \quad (1)$$

where I(x,y) is the image of interest. The family of level sets is defined for all values of λ_i in the range of I and provides a complete, contrast invariant representation of the image [42]. The *principal level sets* are the level sets that contain the basic information required to recover the image. The process of extracting the principal level sets is a non-uniform scalar quantization on the image range. Since the number of principal level sets is less than the total number of levels in the image, the representation is said to be lossy. The detection of the number of principal levels sets and the principal levels themselves has to be a feature-based process in order to preserve person identifiability.

3.2.1 Level Set Decomposition

In order to decompose images into their level sets, a normalization of the images is performed in each subclass. First, the mean of the brightness values of each image in a subclass is equalized. For example, for the Caucasian subclass, the mean of every image is brought 150. Next, the level sets for an image are selected around the fixed mean with displacements of multiples of the standard deviation of the image.

By examining the database, we have determined that the facial images in the database can be represented by seven level sets and recovered with high quality. The λ_i values of the levels sets are chosen at brightness values at {3, 2, 1, 0.5} times the standard deviation less than the mean, at the mean, and at {0.5, 1} times the standard deviation plus the mean. Figure 3.2 shows two database images and the mean image, with their

decomposition into their seven level sets, at respective λ_i values, and enumerated 1 through 7. Note the similarities between the appearances of the features at the same levels of different images. The levels that contain the most common information are the lower (1-2) and higher (6-7) levels. This is due to the fact that the lower levels give a coarse localization of the features, and the higher levels roughly extract the brightest areas of the image, the forehead, cheeks and partially the nose.

On the other hand, the remaining levels contain person-specific contents of the features, such as shape and size. The jaw line information is contained in the second levels sets, the basic mouth template is contained in the third level sets, and the important shading contents around the features are contained in the fourth (around the mouth and nose) and fifth level sets (around the eyes). The principal level sets are therefore chosen to be {2, 3, 4, 5} (Figures 3.1: (a) 2-5, (b) 2-5, (c) 2-5).

3.2.2 Image Reconstruction

An image is recovered from its family of level sets by

$$\hat{I}(x,y) = \max \{\lambda_i : L_i^{(I)}(x,y) = 1\}, \quad (2)$$

where i is the number of level sets and λ_i is the corresponding brightness value at the i^{th} level of the mean image. Figures 3.3.c-d show the reconstructed images of Figures 3.3.a-b from their seven level sets, whereas the reconstructed images from principal levels sets are shown in Figures 3.3.e-f.

Figure 3.2: (a),(b) Examples of cropped database images, (c) the mean image; and their level sets, 1 through 7, in ascending order.

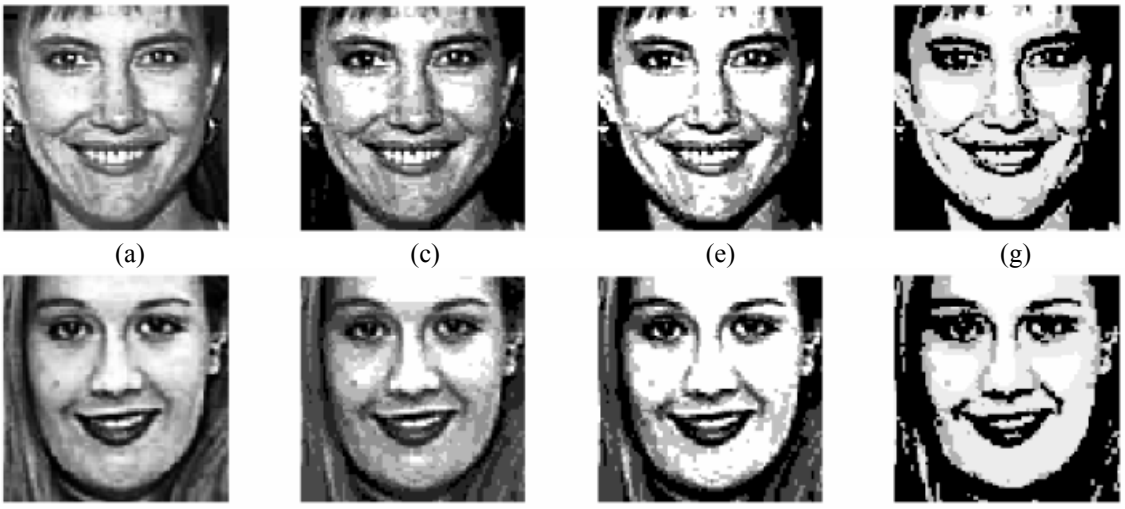


Figure 3.3: (a)-(b) The database images, and the reconstructed images from (c)-(d) seven level sets, (e)-(f) four principal level sets, (g)-(h) four modified principal level sets.

Our goal, recall, is to reduce the amount of data to be transmitted and to therefore reduce the number of principal level sets to represent the image. This goal will be achieved in two stages. First, any excess information in successive level sets that do not contribute to the features will be eliminated and the level sets will be combined. It may be observed that the upper halves of the fourth level sets contain redundant information around the eye regions through the transition from the third through the fifth level sets. Additionally, the lower halves of the fifth region contain shading information caused by illumination around the chin, and is hence not feature based. Thus, the upper half of the fifth is superposed on the fourth level set. This process is depicted in Figure 3.4.

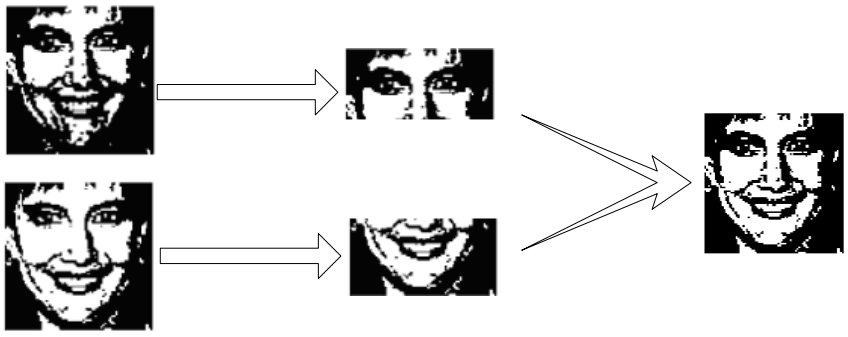


Figure 3.4: Combining the upper and lower halves of the fourth and fifth level sets.

Next, the level sets that share regions with the level sets of the mean image will be locally available at the receiver end. The second level set of a database image may be replaced by an approximate level set attained by a logical OR operation of the second level set extracted from the mean image and the third level set of the database image. This process is shown in Figure 3.5. Finally, the seventh level set of the mean image, which is common to all images, is chosen to add contrast to the reconstructed image. Figure 3.6 presents the modified principal level sets for the database images. The reconstructed images from the new modified principal level sets are shown in Figures 3.3.g-h.

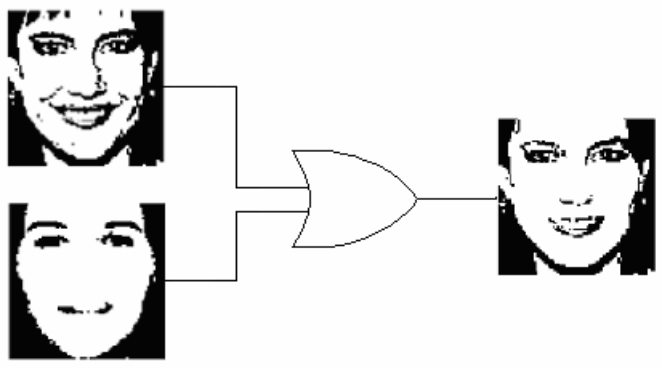


Figure 3.5: Logical ORing of the third level sets of a database image and the mean image to attain the modified level set that will replace the second level set.

Since two of the principal level sets are obtained from the level sets of the mean image, and will be stored at the receiving end, only two level sets are remained to be transmitted for the reconstruction of a database image, namely the second and third modified level sets, given in Figures 3.6.c-f.

Additional examples of level set decomposition of database images and their reconstruction from the modified principal level sets are given in Figure 3.7.

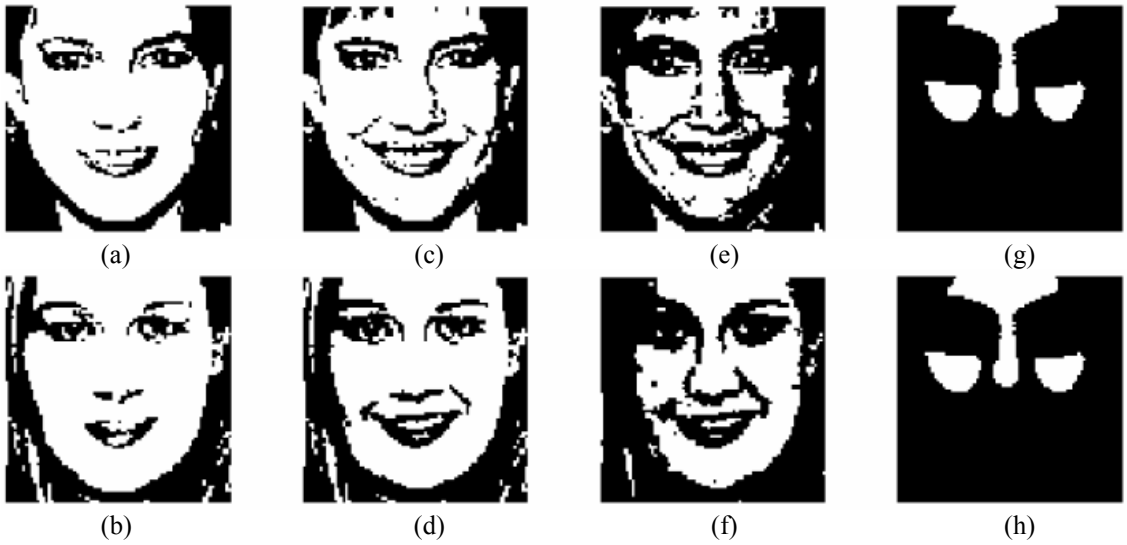


Figure 3.6: The modified principal level sets: (a)-(b) the modified second level sets, obtained via logical ORing the third level sets with the second level set of the mean image, (c)-(d) the third level sets (unmodified), (e)-(f) the modified fourth level sets, obtained via combining the fourth and fifth level sets, (g)-(h) the seventh level set of the mean image.

Figure 3.7: Further examples of database images and their modified level set representations.

3.3 Level Set Compression

3.3.1 Run-length Coding

Run-length coding (RLC) or *variable-to-fixed-length coding* is the most basic lossless compression technique in which a sequence of repeated data values is replaced with the count number or "run" of the value [43]. In other words, instead of coding the individual values, only the lengths of the runs are encoded.

The model that gives rise to run-length coding is the Capon Model [44], a two-state first order Markov model with states white and black. The model states that high compression rates with coding runs are possible if $P(w|w)$ and $P(b|b)$ is close to unity. This is the case for a binary image where white and black runs will be very long, regardless of the value of $P(w)$ or $P(b)$.

For example, in an 8-bit RLC scheme, if we had 180 white pixels followed by 40 black pixels followed by another 230 white pixels, instead of coding the 500 binary numbers individually, we would code the sequence 180, 40, 230, along with an indication of the value of the starting pixel in 25 bits. However, the compression ratio of RLC is context dependent and if the transition probabilities between states are high, then the run-length code may exceed the actual file size.

In an n-bit RLC scheme, if the run length of a value exceeds 2^n-1, then the run length has to be encoded with a multiple of n bits, or it can be expressed in $2n$ bits as:

$$R = (2^n - 1) \times m + t \,, \tag{3}$$

in cases of extremely long run lengths, where m and t are less than 2^n [45].

The bilevel image compression technology used in today's facsimile systems [45, 46] use RLC models followed by static Huffman coding of the run-lengths [47]. The names

"Group 3" and "Group 4", are used sometimes to refer to them, and come from protocols standardized by CCITT (now ITU-T) [48].

3.3.2 Run-length Coding of The Level Sets

The level sets are bilevel images consisting of high repetitions of zeros followed by high repetitions of ones. Our goal is to therefore adapt a scheme similar to facsimile encoding, i.e., a RLC scheme followed by arithmetic coding [49], a minimum redundancy coding algorithm, which is superior to Huffman coding in small alphabets. An arithmetic coding example is provided in the appendix.

The optimum bit length for RLC of the level sets was found to be $n = 5$. Hence, the possible run-lengths are between 0 and 31, and the observed distribution of the probabilities of the run-lengths is almost uniform. Both Huffman and arithmetic coding, however, require alphabets with skewed probabilities, since the key point in both schemes is to code frequent symbols with fewer bits [47, 49]. Therefore, further compression of the run-length code with these algorithms is negligibly small. The compression results are provided in *Chapter 5*.

Chapter 4

Wireless Network Communication and Transmission of Facial Images

The wireless network infrastructure of North Carolina Criminal Justice Information Network (CJIN) is managed and operated by the State Highway Patrol (NCSHP). This network serves to send information to mobile trooper and police cars, which are equipped with portable computers and radio modems which function under Windows 95 operating systems. With over 275 agencies, there are almost 8000 officers registered to the CJIN wireless network. During peak usage, the approximate number of users in the network is 5000, with an average of 2500 daily/evening users - dropping to 2000 during early morning hours.

The radio modem in the mobile vehicle, communicates with the nearest base tower at a low rate 19.2kbps shared channel. Actual throughput on the channel is approximately 10kbps - the actual rate varying due to range and S/N issues. The radio tower is connected to the state network via a 14.4kbps/56K line through radio network controllers and radio switches. The overall transmission schematic is shown in Figure 4.1.

The end-to-end transmission of the facial images is a TCP/IP (Transmission Control Protocol/Internet Protocol) client-server application. A request from the client for a driver's license image is sent to the server and is serviced at the database, where a compressed image is transmitted back to the client. At the client side, the decompression

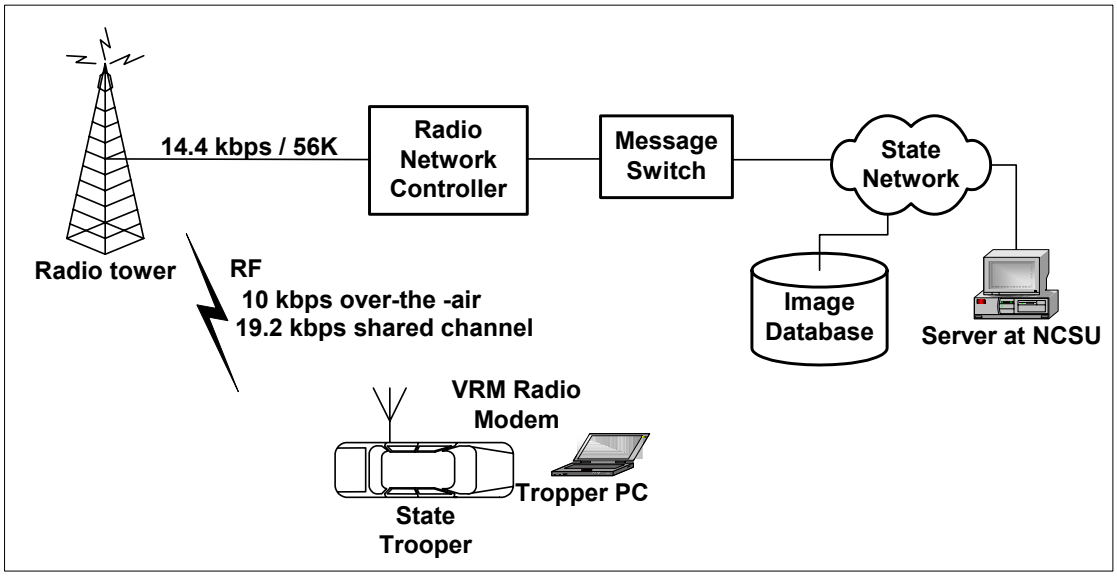


Figure 4.1: The end-to-end transmission schematic.

application is run and the image is displayed to the user. The user interfaces of both the client and server applications are created using Microsoft Foundation Classes (MFC) [50, 51].

4.1 Windows Sockets

The basic object used by applications for most network communications is called a *socket.* A socket is an abstract representation of a communication endpoint, which allows the associated application to send and receive packets of data across the network. It should be pointed out that sockets are purely software objects, and they are not protocols but application program interfaces (API) between an application and a network protocol. Sockets API is a high level abstraction of the processes that are required to execute a given function, which takes away the need for the programmer to have any knowledge on how the network behaves [52].

Sockets were first developed on UNIX operating at the University of California Berkeley in the early 1980's with the funding of ARPA [53].

Windows Sockets or *WinSock* is a network programming interface specifically for Microsoft Windows. Leffler et al. [54] describe the *Berkeley Software Distribution* socket system from which Windows Sockets is derived. The Windows Socket specification, now at version 2.0, was developed as an open networking standard by a large group of individuals and corporations in the TCP/IP community and is available in the Win32 SDK.

There are two methods in Windows Sockets for the operation of handling the transmission of data packets across the network, namely, *datagram* and *stream sockets*. Upon creation, each socket has to be defined as to which method it uses.

- *Datagram (UDP) Sockets* operate on a connectionless principle, that is, no explicit connection is established between the two endpoints. As a result, the data may not arrive in sequence, be duplicated, or not arrive at all. Thus, the programmer is responsible for managing sequencing and reliability. The best applications for this socket type are when the data itself contains the information as to its sequence, such as broadcasting the system time of a network to keep it synchronized.
- *Stream (TCP) Sockets*, on the other hand, require an explicit connection between the two endpoints for communication to occur. With this type of connection the sequence and arrival of unduplicated data is guaranteed. Furthermore, stream sockets are well suited to handling large amounts of data.

For our specific application, stream (TCP) sockets will be used, since an explicit connection is required between the server and the client for real-time transmission.

4.2.1 The Client-Server Model

Prior to establishing a stream socket connection, the server has to create a "listening socket" at a port, which will be allocated to a specific application and where the server will be waiting for a connection request from a client to use the application. Only one application may be listening on any specific port on a single computer. Although numerous applications can listen for connection requests on a single computer at the same time, each of these applications must listen on a different port. If the maximum capacity for the application is reached, the listening socket may queue pending requests.

A client, on the other hand, needs explicit information about the internet protocol (IP) address of the server and the port number allocated to the application he wants to run. The client creates a socket with the server address and application port number, that is, sends a connection request to the server. If the capacity limit is not reached, the listening socket creates another socket, the "connection socket", which is connected to the client socket. All further communication between the client and the server is made through the client socket and the connection socket of the server. The listening socket meanwhile continues listening at the port for other client requests.

Once the connection is established, the client sends the name of the person whose driver's license photograph is of interest to the server. The server locates and compresses the image and transfers it to the client along with the size of the file. As soon as the file is received completely, the client closes the connection and releases the connection socket of the server. At the client side, the received file is decompressed and the recovered driver's license image is displayed to the user.

4.2 MFC Programming

The client application program requires a user interface to a start connection, to send file requests, and to display images. Creating Windows applications of such sort is feasible with MFC programming using Visual C++. A user-friendly interface can be created via MFC, which can also handle and display image objects. Moreover, MFC supplies models for writing network communication programs with Windows Sockets, embodied in two MFC classes: CAsyncSocket and CSocket. Further details on these classes may be found in [50, 51].

The server and client user interfaces are shown and the overall transmission is summarized in Figure 4.2.

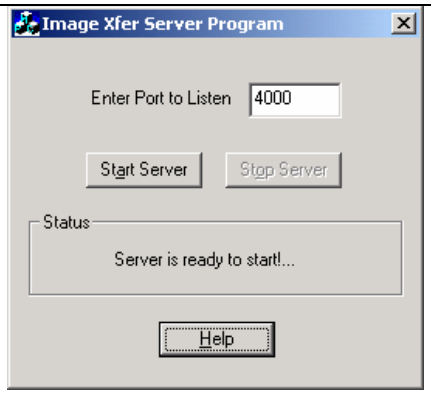
Server	Client
 (1) Initialization of the server program; the user enters the port number at which the application will be listening and clicks on the "Start Server" button.	

Figure 4.2: The sequence of Server-Client connection and image transmission.

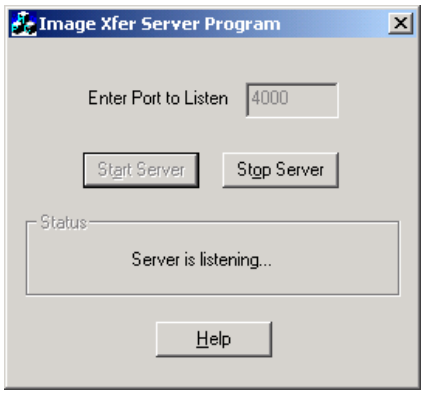
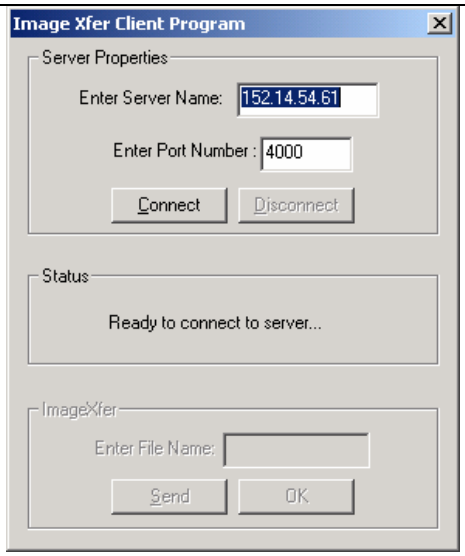
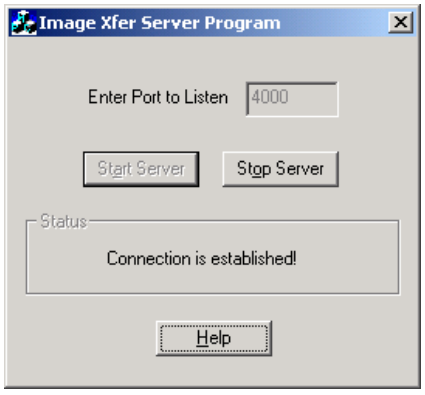
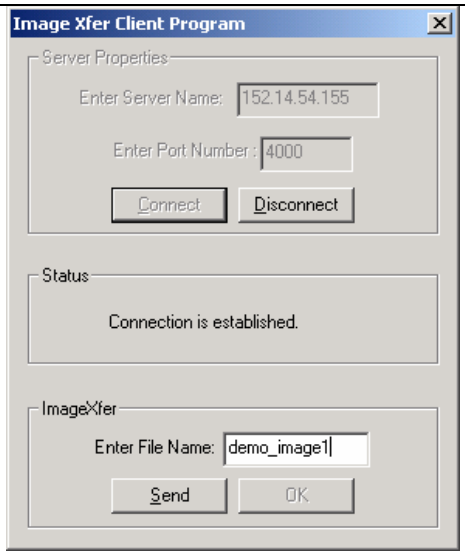
Server	Client
	
(2) The server creates the listening socket at the port number and waits for connection requests.	(3) Initialization of the client program; the user enters the server IP address and the port number of the application, creating a socket, and sends the connection request by pressing the "Connect" button.
	
(4) The server accepts the request, and creates a new socket for connection.	(5) Once the connection is established, the user enters the name of the image for transmission.

Figure 4.2: continued.

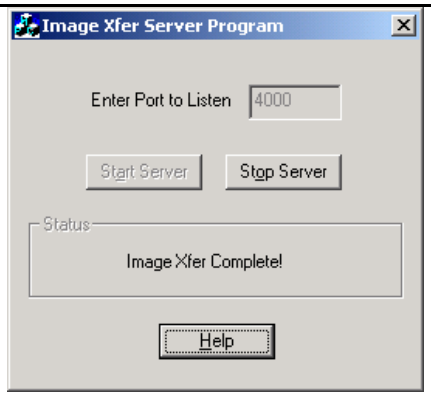
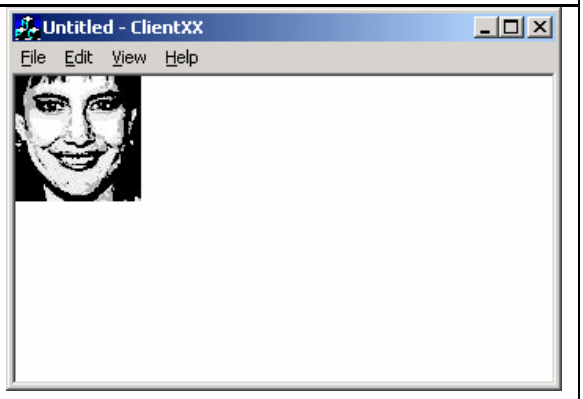
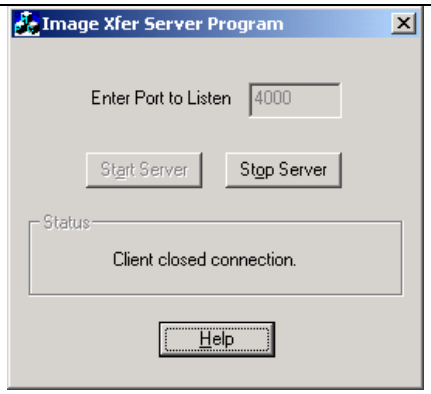
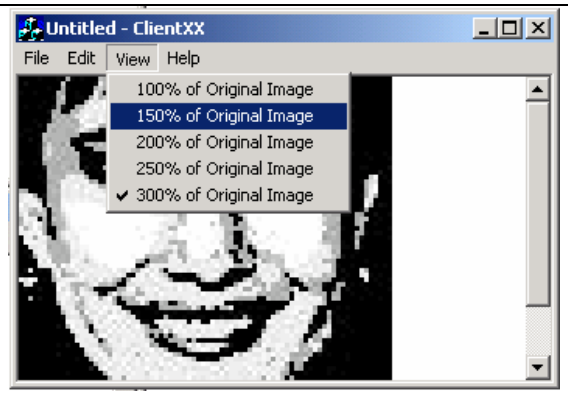
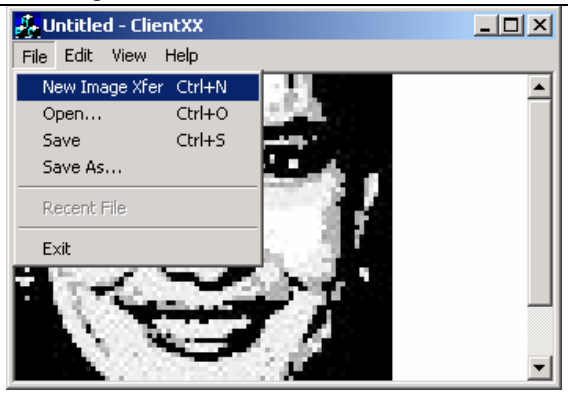
Server	Client
	
(6) The server locates, compresses and transmits the image to the client.	(7) The received image file is decompressed and is displayed to the user. The connection is closed as soon as the file is received.
	
(8) The server is notified that the client end closed the connection and frees its connected socket.	(9) The user may change the viewing size of the image via the View menu.
	
	(10) The user may send a new transmission request, open an existing image, or save the current image via the File menu.

Figure 4.2: continued.

Chapter 5

Experimental Results and Conclusion

5.1 Facial Feature Extraction

In a sample set of database images consisting of 100 Caucasians with no auxiliary features, 10 Caucasians with eyeglasses, 10 Caucasians with mustaches, and 40 African-Americans, the proposed algorithm has successfully classified each image into its respective class.

Within the sample set of 100 Caucasians, we have accurately detected the eye and mouth regions using the extracted ravines, obtained by the proposed feature extraction algorithm.

In the presence of mustaches, an estimation of the mouth region with respect to the mustache positioning is made, and was verified to be adequate with the experimental results. Next, in the presence of eyeglasses, results showed that further search for the eye regions should not be conducted due to cases of blocked eyes caused by the frame or reflections from the glass surface. The centroids of the lobes of the eyeglasses were extracted instead, which in turn were used to align the images.

In the case of African-Americans, the curvature of the valley of mouth openings was observed not to be as significant as in Caucasians. This caused a decrease in the detection rate of the mouth regions to 92.5%. However, it was also observed the curvatures of the nose-base regions were more useful when compared to the Caucasian group. The nose

regions may, thus, be preferably extracted (rather than the mouth regions) for alignment to achieve total accuracy within this class.

5.2 Compression Results and Transmission Statistics

The compression and transmission experimental results of 40 typical images from the group of Caucasians are presented in Table 5.1. The original images are 360x300x24 bits and are stored in JPEG format. The average file size of the samples is 9.94 Kbytes. The average compression ratio of the implemented algorithm was found to be 92.02%. Hence, the average size of the file that has to be transferred is reduced to 789.375 Bytes. The resulting images are 80x80 pixels yielding an average compression bit rate of 0.9867 bits/pixel. An important observation to make is that the compression rate attained after run-length coding is 91.78%. This implies that the probability distribution of the run-lengths is almost uniform. Further arithmetic coding the files, therefore, provides a compression ratio on the average of only 2.24%.

The effect of the overall reduction in the file sizes is reflected by the achieved transmission statistics. The transmission statistics were taken on different time slots of different days. In Table 5.1 the average of these statistics for each image is presented. While the average transmission time of original database images was found to be 28.53 seconds, the average transmission time of the compressed images was found to be 4.64 seconds. This result illustrates a 6x improvement in the transmission time. The total compression and decompression time was approximately 2.5 seconds. The improvement in the overall process is, hence, approximately 4-fold.

Original Size (in KBytes)	Compressed Size (in Bytes)	Compression Ratio	Average Transmission Time of Original Image	Average Transmission Time of Compressed Image	Improvement in Overall Transmission Time
5.92	774	87.23%	21.50 sec	4.50 sec	3.07
7.30	799	89.31%	23.25 sec	4.75 sec	3.21
7.79	805	89.91%	23.50 sec	4.50 sec	3.36
7.91	864	89.33%	24.25 sec	5.00 sec	3.23
8.19	834	90.06%	24.25 sec	4.00 sec	3.73
8.38	780	90.91%	24.25 sec	4.50 sec	3.46
8.58	788	91.03%	24.75 sec	5.00 sec	3.30
8.66	611	93.11%	25.00 sec	4.25 sec	3.70
8.70	775	91.30%	25.25 sec	4.75 sec	3.48
8.89	941	89.66%	25.25 sec	4.75 sec	3.48
8.93	989	89.18%	26.00 sec	4.75 sec	3.59
8.99	836	90.92%	26.00 sec	4.25 sec	3.85
9.16	795	91.52%	26.00 sec	4.50 sec	3.71
9.16	885	90.56%	26.25 sec	5.00 sec	3.50
9.19	786	91.65%	26.75 sec	4.50 sec	3.82
9.23	625	93.39%	27.00 sec	4.25 sec	4.00
9.31	718	92.47%	27.00 sec	4.50 sec	3.86
9.39	595	93.81%	27.25 sec	4.00 sec	4.19
9.40	828	91.40%	27.75 sec	5.50 sec	3.47
9.41	829	91.40%	28.00 sec	4.50 sec	4.00
9.53	713	92.69%	28.25 sec	4.50 sec	4.04
9.62	663	93.27%	28.75 sec	4.00 sec	4.42
9.68	611	93.84%	29.00 sec	4.50 sec	4.14
9.74	1024	89.73%	29.00 sec	5.75 sec	3.51
10.60	720	93.37%	29.00 sec	4.50 sec	4.14
10.70	811	92.60%	29.00 sec	4.50 sec	4.14
10.80	833	92.47%	29.25 sec	4.50 sec	4.17
10.80	742	93.29%	29.25 sec	4.50 sec	4.17
11.10	813	92.85%	29.25 sec	5.00 sec	3.90
11.20	750	93.46%	30.00 sec	4.75 sec	4.14
11.30	601	94.81%	30.75 sec	4.00 sec	4.73
11.60	801	93.26%	30.75 sec	5.00 sec	4.10
11.60	838	92.95%	31.50 sec	4.75 sec	4.34
11.70	941	92.16%	31.50 sec	5.25 sec	4.06
11.70	733	93.88%	31.75 sec	4.75 sec	4.38
11.90	850	93.02%	32.00 sec	5.00 sec	4.27
12.00	656	94.66%	32.00 sec	4.25 sec	4.74
12.50	894	93.02%	35.75 sec	5.00 sec	4.77
12.80	892	93.19%	36.50 sec	5.00 sec	4.87
14.30	832	94.32%	48.75 sec	4.50 sec	6.96

Table 5.1: Compression results and improvements in overall transmission times of 40 typical facial images from the driver's license database.

5.3 Summary and Conclusion

Specialized techniques for compression of facial images are essential for their transmission over wireless channels. In this thesis, we presented a feature-based compression technique, relying on the fact that at some level, all faces have a common template that need not be transmitted.

First, we attempted to extract the facial features with a methodology based on the topological nature of the human face. Ravines, by definition resemble the valley regions of a surface, and thus, were used to detect the eyes and the mouth, which correspond to deep valleys on the face surface. For every dataset image, the extracted triangles whose vertices correspond to the centroids of the eye and mouth regions were aligned via resizing. The facial regions were extracted from the images so that only the necessary information for identifiability is transmitted.

It was observed from the database that some images of people exhibit more common features than others. The database, therefore, was divided into subclasses depicting similar features. Next, the means of each subclass were attained and analyzed within the subclass via level set decomposition. The level sets containing important facial features for person identification and the level sets sharing common information were chosen for the reconstruction of the facial images. The elimination of the redundant level sets and the storage of the common level sets at the destination form the bases of our compression methodology. The principal level sets to be transmitted are compressed via run-length coding and arithmetic coding. The methodology offered high compression rates while sustaining person identifiability. The high compression rates were reflected on the transmission times of the compressed image files.

Finally, the transmission was accomplished via a TCP/IP client-server model functioning with Windows Sockets. The user-interfaces were implemented using Visual C++, enabling a user-friendly environment, the usage of Windows Sockets and the display of images.

In summary, we have presented a methodology that provides a 4x improvement of the transmission time of driver's license images over narrowband wireless channels.

The future work should focus on the following three aspects: increasing the compression rate, improving the image quality, and adding a verification stage to the facial feature algorithm. Further compression may be achieved by means of adaptive two-dimensional run-length coding models rather than coding in one dimension. If adequate compression rates are achieved, the quality of the images may be improved by forward quantization, which suggests the division of the image into blocks and adding the minimum and maximum brightness values of each block as side information, and as a reference for improved recovery [44]. In addition, if the quality of a recovered image is poor, the client may be given the user rights to request the transmission of additional level sets. The facial feature extraction is a crucial stage for the methodology to achieve its goal. A verification stage combining the proposed technique with other methods of feature extraction proposed in the literature would definitely yield improved accuracy.

Bibliography

[1] H.D. Ellis, "Introduction to aspects of face processing: Ten questions in need of answers", in *Aspects of Face Processing*, H. D. Ellis, M. Jeeves, F. Newcombe, and A. Young Eds. Dordrecht:Nijhoff, 1986, pp.3-13.

[2] ____, "The role of the right hemisphere in face perception", in *Function of the Right Cerebral Hemisphere*, A. W. Young Ed. London: Academic, 1983.

[3] R. Bruyer, *The Neuropsychology of face perception and facial expression*, Hillsdale, N.J. : L. Erlbaum Associates, 1986.

[4] V. Bruce, *Recognizing Faces*, London: Erlbaum Associates, 1988.

[5] V. Bruce, A. Young, and A.W. Young, *In the Eye of the Beholder: The Science of Face Perception*, Oxford University Press, 1998.

[6] Young, *Face and Mind*, Oxford University Press, 1998.

[7] R. Brunelli, T. Poggio, "Face Recognition: Features vs. Templates", *IEEE Transactions on Patter Analysis and Machine Intelligence*, vol. 15, pp. 1042-1052, Oct. 1993.

[8] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey", *Proceedings of IEEE*, vol. 83, pp. 705-740, May 1995.

[9] L. Sirovich, M. Kirby, "Low-dimensional procedure for the charaterization of human faces", *Journal of the Optical Society of America*, vol.4, pp. 519-524, March 1987.

[10] M. Kirby, L. Sirovich, "Application of the Karhunen-Loève Procedure for the characterization of human faces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12, no.1, pp.103-108, Jan. 1990.

[11] S. Watanabe, "Karhunen-Loève expansion and factor analysis theoretical remarks and applications", *Proceedings of the 4th Prague Conference on Information Theory*, 1965.

[12] M. Turk, A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, no.1, pp.71-86, 1991.

[13] P.S. Penev, L. Sirovich, "The global dimensionality of face space", *Proc. 4th IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 264-270, 2000.

[14] P.S. Penev, J.J. Atick, “Local feature analysis: A general statistical theory of object representation”, *Network: Computation in Neural Systems*, 7(3):477-500, 1996.

[15] D. Shah, S. Marshall, “Statistical coding method for facial features”, *IEE Proc. Vis. Image Signal Process.*, vol. 145, no. 3, pp. 187-192, June 1998.

[16] J.H. Hu, R.S. Wang, and Y. Wang, “Compression of personal identification pictures using vector quantization with facial feature correction”, *Optical Engineering*, vol. 35, no.1, pp.198-203, Jan.1996.

[17] L. Mygatt, “Summus Photo Solutions”, *White Paper 2001:1*, Summus, Inc., 2001.

[18] T. Kanade, *Computer Recognition of Human faces,* Basel and Stuttgart: Birkhauser, 1997.

[19] R.S. Wang, Y. Wang, “Facial feature extraction and tracking in video sequences”, *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, Princeton, New Jersey, pp. 233-238, June 1997.

[20] M. Nixon, “Eye spacing measurement for facial recognition”, *SPIE Proc.*, vol.575, pp.279-285, 1985.

[21] A. Yuille, D. Cohen, and P. Hallinan, “Facial feature extraction from faces using deformable templates”, *Proc. IEEE Computer Soc. Conf. On Computer Vision and Pattern Recognition*, pp. 104-109, 1989.

[22] A.W.M. Kass, D. Terpoulos, “Snakes: Active contour models”, *Proc. International Conference on Computer Vision*, pp/ 259-269, 1987.

[23] Y. Wang and O. Lee, "Active Mesh --- A Feature Seeking and Tracking Image Sequence Representation Scheme," *IEEE Trans. Image Processing, Special issue on image sequence coding*, vol. 3, pp. 610--624, Sept. 1994.

[24] C.H. Lin, J.L. Wu, “Automatic facial feature extraction by genetic algorithms”, *IEE Transactions on Image Processing*, vol. 8, no. 6, June 1999.

[25] G.G. Yen, N. Nithianandan, “Facial feature extraction using genetic algorithm”, *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, pp. 1895-1900, 2002.

[26] T.C. Chang, T.S. Huang, and C. Novak, “Facial feature extraction from color images”, *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 2, pp. 39-43, Oct 1994.

[27] M. Ikeda, H. Ebine, O. Nakamura, "Extraction of faces of more that one person from natural background for personal identification", *Canadian Conference on Electrical and Computer Engineering,* vol.1, pp. 323-328, 2001.

[28] B.S. Manjunath, R. Chellappa, C. Malsburg, "A feature based approach to face recognition", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 373 –378, June 1992.

[29] Y. Tian, T. Kanade, and J.F. Cohn, "Evaluation of Gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity", *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 218 –223, May 2002.

[30] D. Taubman, M.W. Marcellin, *Jpeg2000:Image Compression Fundamentals, Standards, and, Practice,* Kluwer Academic Publishers, Boston, November 2001.

[31] F. Morgan, *Riemannian Geometry*, A K Peters Ltd, Wellesley, MA, 1998.

[32] Gray, A., *Modern Differential Geometry of Curves and Surfaces*, CRC Press, Boca Raton, Florida, 1998.

[33] A.B. Hamza, H. Krim, "A topological variational model for image singularities", *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, vol.1, pp. 369-372, September 2002.

[34] E.V. Anoshkina, A.G. Belyaev, R. Huang, T.L. Kunii, O.G. Okunev, S. Takahashi, "Hierarchic shape description via singularity and multiscaling", *Proceedings of IEEE Computer Software and Applications Conference*, 1994.

[35] R. Huang, T.L. Kunii, "Parallel algorithms for extracting ridges and ravines", *Proceedings of the First Aizu International Symposium on Parallel Algorithms/Architecture Synthesis*, March 1995.

[36] P.W.M Tsang, W.H. Tsang, "Edge detection on object color", *Proceedings of International Conference on Image Processing*, vol. 3, pp. 1049-1052, Sep 1996.

[37] W.C. Huang, C.H. Wu, "Adaptive color image processing and recognition for varying backgrounds and illumination conditions", *IEEE Transactions on Industrial Electronics*, vol. 45, no. 2, pp. 351-357, April 1998.

[38] http://www.mathworks.com/access/helpdesk/help/toolbox/images/color11.shtml

[39] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Addison-Wesley Pub. Co., Jan 2002.

[40] O. Monga, N. Armande, and P. Montesinos, "Thin nets and crest lines: application to satellite data and medical images", *Computer Vision and Image Understanding*, vol.67, pp. 285-295, September 1997.

[41] A. Gunduz, H. Krim, "Compression and Transmission of Facial Images Over Very Narrowband Channels", *to be published in Proc. of Int. Conf. Acoustics, Speech and Signal Processing 2003.*

[42] C. Kervrann, M. Hoebeke, and A. Trubuil, "A level line selection approach for object boundary estimation", *Proc. IEEE Int. Conf. on Computer Vision*, Greece, vol.2, pp. 963-968, September 1999.

[43] http://www.rasip.fer.hr/research/compress/algorithms/fund/rl/

[44] J. Capon, "A probabilistic model for run-length coding of pictures", IRE Transactions on Information Theory, pp. 157-163, December 1959.

[45] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, San Francisco, California, p. 123, 1996.

[46] R.B. Arps, "Comparison of international standards for lossless still image compression", *Proceedings of IEEE*, vol. 82, no. 6, June 1994.

[47] D.A. Huffman, "A method for reconstruction of minimum redundancy codes", *Proceedings of IRE*, vol. 40, pp. 1098-1101, 1951.

[48] CCITT Blue Book, Terminal Equipment and Protocols for Telematic Services-Recommendations t.0-t.63, Geneva: International Telecommunication Union, 1989.

[49] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression", *Communications of the ACM*, vol. 30, no. 6, June 1987.

[50] D. White, K. Scribner, and E. Olafsen et al., *MFC Programming with Visual C++*, Sams Publishing, pp. 851-884, 1999.

[51] K. Gregory, *Special Edition Using Visual C++*, Que Publishing, pp. 427-429, 1998.

[52] D. E. Comer, D. L. Stevens, *Internetworking with TCP/IP, Volume 3: Client-Server Programming and Applications, Windows Sockets Version,* Prentice-Hall Inc., Upper Saddke River, New Jersey, 1997.

[53] V. Cerf, "A History of the ARPANET", *ConneXions*, *The InteroperabilityReport*, Foster City, California, 1989.

[54] S. Leffler, M. McKusick, M. Karels, and J. Quarterman, *The Design and Implementation of the 4.3BSD UNIX Operating System*, Addison-Wesley, Reading, and Massachusetts.

Appendix

A. Auxiliary Facial Feature Extraction

A.1. Eyeglass Detection

When the algorithm is applied to a facial image consisting of eyeglasses, the ravines of the extracted box, detect the frame of the eyeglasses. Moreover, when morphological filling is applied, the eyeglass region is extracted as shown in Figure A.1.b. In the search of possible eye regions stage, since this region is large, it is thresholded, and no eye pairs are captured. The algorithm, then searches for possible eyeglass regions. The centroids of the eyeglasses may be localized by means of Hough transform [20], for alignment within the subclass.

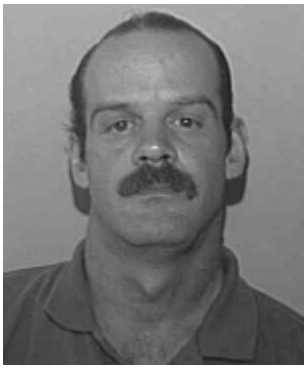
(a) (b)

Figure A.1: (a) A facial image with eyeglasses, (b) morphologically filled ravines.

A.2. Mustache Detection

After the eye regions extraction in a facial image with a mustache, the mustache region is revealed in the mouth extraction stage (see Figure A.2). The extraction of a large region

in the mouth box is perceived as the existence of a mustache region and the localization of the mouth region is estimated with respect to the mustache region.



(a)

(b)

Figure A.2: (a) A facial image with mustache, (b) morphologically filled ravines.

B. Arithmetic Coding

Arithmetic coding is a minimum redundancy coding technique, which dispenses with the restriction that each symbol must translate into an integral number of bits, therefore compresses data more efficiently. Compression is achieved by coding the more probable symbols in fewer bits than the less probable ones.

In arithmetic coding, a message is represented by an interval of real numbers between 0 and 1, i.e. [0,1). Initially, the interval [0,1) is divided into subintervals of all possible symbols to appear within the message. The size of each subinterval is proportional to the frequency at which symbols appear in the message. At every new symbol, the interval is replaced by the subinterval corresponding to that symbol. The new interval is similarly divided into subintervals. As the message becomes longer, the required interval to represent it becomes smaller, and the number of bits required specifying that interval grows. The more likely symbols reduce the range less than the unlikely symbols, and hence add fewer bits to the message.

Let us consider an example where we have an alphabet composed of the symbols {a, b, c, d}, with the following probabilities:

Symbol	Probability	Interval
a	0.2	[0, 0.2)
b	0.3	[0.2, 0.5)
c	0.1	[0.5,0.6)
d	0.4	[0.6, 1)

Table B.1: An example alphabet with the respective the probabilities and coding intervals of each symbol.

The interval [0,1) is divided into subintervals whose lengths are proportional to the frequency of the symbols as shown in Table B.1. Let us code the word “add”.

The first symbol that needs to be coded is “a”. The initial subinterval corresponding to the symbol “a” is [0, 0.2). This subinterval is divided into subintervals in the same manner (see Table B.2).

Symbol	Probability	New Interval
a	0.2	[0, 0.04)
b	0.3	[0.04, 0.1)
c	0.1	[0.1,0.102)
d	0.4	[0.102, 0.2)

Table B.2: The updated coding intervals of the symbols after the symbol “a”.

The next symbol in the word is “d”. The interval is thus replaced by [0.102, 0.2). The following intervals are attained, once this interval is divided:

Symbol	Probability	Interval
a	0.2	[0.102, 0.1216)
b	0.3	[0.1216, 0.151)
c	0.1	[0.151, 0.1608)
d	0.4	[0.1608, 0.2)

Table B.3: The updated coding intervals of the symbols after the code “ad”.

For the following symbol “d”, we have the interval [0.1608, 0.2). Hence, we can code the word “add” with any real number in the interval, for example 0.1825. This decimal number is converted to binary, 0.0011, and the digits after the decimal point constitute the message, 0011, which can be coded in 4 bits. The Huffman code of the same word “add” would be coded as “a”=010, “d”=1, “d”=1, in 5 bits.