Automatic Facial Expression Recognition (AFER)

Developed by

Sadia Arshid
Maham Javed

Supervised by

Dr. Syed Afaq Hussain

Department of Computer Science
Faculty of Applied Sciences
International Islamic University, Islamabad.
(2006)

بسم الله الرحمن الرحيم

In The Name of

ALLAH ALMIGHTY

The Most Merciful The Most Beneficent

International Islamic University, Islamabad
Faculty of Applied Sciences
Department Of Computer Sciences

Date: 12-8-2006

Final Approval

It is certified that we have read the project report, entitled as "**Automatic Facial Expression Recognition (AFER)**" submitted by Sadia Arshid and Maham Javed. It is our judgment that this project is of standard to warrant its acceptance by the International Islamic University, Islamabad, for the MS Degree of Computer Science.

Project Evaluation Committee

External Examiner

Dr. Mehboob Yasin,
Professor,
Comsats Institute of Information Technology,
Islamabad .

Internal Examiner

Mr. Asim Munir,
Assistant Professor,
International Islamic University,
Islamabad .

Supervisor

Dr. S. Afaq Hussain,
Head of Telecom Department,
Head of Computer Engineering,
International Islamic University,
Islamabad .

A dissertation submitted to the Department of Computer Science,
Faculty of Applied Sciences, International Islamic University, Islamabad, Pakistan,
as a partial fulfillment of the requirements for the award of the degree of

MS in Computer Science

To
The Holiest man Ever Born,
PROPHET MUHAMMAD (PEACE BE UPON HIM)
&
To
OUR PARENTS
We are most indebted to our parents, whose affection has been the source of encouragement for us, and whose prayers have always been key to our success.
&
Our
PRECIOUS FRIENDSHIP
that has made us laugh, held us when we cried
and always, always, always
be among us

Declaration

We hereby declare and affirm that this software neither as a whole nor as a part thereof has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts, made under the sincere guidance of our teachers. If any part of this project is proven to be copied out or found to be a reproduction of some other, we shall stand by the consequences.

No portion of the work presented in this report has been submitted in support of an application for other degree or qualification of this or any other University or Institute of learning.

Sadia Arshid
232-CS/MSCS/F04
Maham Javed
225-CS/MSCS/F04

Acknowledgement

We bestow all praises to, acclamation and appreciation to Almighty Allah, The Most Merciful and compassionate, The Most Gracious and Beneficent, Whose bounteous blessings enabled us to pursue and perceive higher ideals of life, who bestowed us good health, courage and knowledge to carry out and complete our work. Special thanks to His Holy Prophet Muhammad (SAW) who enabled us to recognize our Lord and Creator and brought us the real source of knowledge from Allah, the Qur'an, and who is the role model for us in every aspect of life.

We consider it a proud privileged to express our deepest gratitude and deep sense obligation to our reverend supervisor **Dr. Afaq Hussain** who kept our morale high by his suggestions and appreciation. His motivation leads us to this success without his sincere and cooperative nature and precious guidance; we could never have been able to complete this task.

It will not be out of place to express our profound admiration for our teacher **Mr. Asim Munir** for his dedication, inspiring attitude, untiring help and kind behavior through out the project efforts and presentation of this manuscript. He did a lot of effort for our success.

Finally we must mention that it was mainly due to our family's moral support and financial help during my entire academic career that enabled us to complete our work dedicatedly. We once again would like to admit that we owe all our achievements to our most loving parents, who mean most to us, for their prayers are more precious then any treasure on the earth.

In the end we are also thankful to our truly, sincere and most loving brothers, sisters , friends, and class fellows who mean the most to us, and whose prayers have always been a source of determination for us.

Sadia Arshid
232-CS/MSCS/F04
Maham Javed
225-CS/MSCS/F04

PROJECT IN BRIEF

Project Title:	Automatic Face Expression Recognition (AFER)
Organization	International Islamic University, Islamabad
Under Taken By:	Sadia Arshid Reg. No# 232/CS/MSCS/F04 Maham Javed Reg. No# 225/CS/MSCS/F04
Supervised By:	Dr.Syed Afaq Hussain Head of Department of Telecommunication International Islamic University, Islamabad.
Starting Date:	November, 2005.
End Date:	June, 2006.
Tools used:	Matlab 7.
System Used	Pentium IV

ABSTRACT

Now a days one of the major areas of research is imitating human behaviour in artificially intelligent machines. Emotions play a vital role in depicting human behaviour and study of emotions comes under the affective computing which includes recognizing, understanding and to some extent exhibiting emotions. Emotions are majorly revealed through expressions. Our area of research is detecting and recognising expressions.

In this project we have developed a system for automatic detection of facial expressions. The work has been done on the static images on which little work has been done as it is difficult to work with static images then on video streams in which optical information is helpful in determining expression. Hue component in HSV colour space is use for determining skin values in face localization. We have done purely color based feature localization which is not done before. Multi detectors are used for feature extraction. Emphasis is given on the local statistics of the features for their enhancement and control point detection.

The statistical analysis of the features control point in comparison with the neutral results in expression recognition. The system gives accuracy of 88%.

Table of Contents

1 Introduction 1

1.2 Emotions 1
1.3 Face & Emotions 1
1.4 Role of Facial Expression in Emotion Recognition 2
1.5 Perception of Face Expressions 2
1.6 Facial Expression Analysis 2
1.6.1 Image acquisition and preprocessing 3
1.6.2 Face localization and detection 3
1.6.3 Feature extraction and Analysis 3
1.6.4 Expression recognition 3
1.7 Color Spaces Used For Skin Modeling 4
1.7.1 Color Spaces 4

2 General Image Processing Techniques 8

2.1 Morphological Operations 8
2.1.1 Dilation 8
2.1.2 Erosion 9
2.1.3 Opening 9
2.1.4 Closing 10
2.1.5 Boundary Extraction 10
2.1.6 Region filling 10
2.2 Filters 11
2.2.1 Smoothing Filters 11
2.2.2 Sharpening Filters & Edge Detectors 12
2.3 Integral Projections 15

3. Literature Survey 16

3.1 Previous work For Face Detection 16
3.2 Recent Research on Facial Expression Recognition 18
3.3 Problem Statement 24

4. Proposed System 25

4.1 Major Modules & Architectural Diagram 25
4.2 Image Acquisition 26
4.3 Face Localization 26
4.3.1 Color Space Conversion 26
4.3.2 Face Segmentation 29
4.3.3 Face Rotation 29

4.3.4 Face Cropping & Rescaling....30
4.4 Face zone division....31
4.5 Feature Extraction....31
4.5.1 Eye Localization....31
4.5.2 Eyebrows Detection....33
4.5.3 Eye Detection....34
4.5.4 Nose Estimation....35
4.5.5 Lip localization and detection....35
4.5.6 Analysis and Expression Recognition....37

5. Results....41
5.1 Experimental Results....41
5.2 Analysis of Results....48

6. Conclusion & Future enhancements....50
6.1 Conclusion....50
6.2 Future Enhancements....53

Appendix....

References....

Publications....

Chapter 1

Introduction

1. Introduction

In the last 30 years one of the most interesting areas of research is building machines that would complement human life with the help of artificial intelligence. This area is full of different challenges and one among them is to imitate human vision. Human vision is one out of many areas that want to understand the process of human functionality and copy that process with intention to complement human life with intelligent machines. For better human-computer interaction it is necessary for the machine to see people. This encouraged new researches with goals to enable a computer to see people, recognizes them and interprets their emotions like gestures, expressions and behavior. Affective computing is a young field of research dealing with the issues regarding emotions and computers.

Affective Computing was introduced by Picard to refer to all "computing that relates to, arises from, or deliberately influences emotions". According to Rosalin Picard (Picard, 1998), if we want computers to be genuinely intelligent and to interact naturally with us, we must give computers the ability to recognize, understand, and even to have and express emotions. Emotions themselves are very human matter, of which there is no clear theory or understanding. The necessary background for affective computing is the knowledge on emotions and their role in human behavior and cognitive processes.

Affective computing consists of four related areas. For communication, computers can both recognize and express emotion. Emotions can be expressed without really having them, like an actor playing a role. Having emotions is a separate, but very profound question. Last, computers could be able to have emotional intelligence. Lately very fashionable in human psychology, it deals with reasoning and understanding of emotions.

1.2 Emotions

What are emotions; is largely remains an open question. Some define it as the physiological changes caused in our body, while the others treat it as purely intellectual thought-process. Emotions are closely related to perception. We understand each other's emotions through visual, auditory and tactile senses.

1.3 Face & Emotions

Faces are the center of human-human communication, and have been object of analysis for centuries. It is evident, that the face conveys to humans such a wealth of social signals, and humans are expert at reading them. They tell us who is the person in front of us or help us to guess features that are interesting for social interaction such as gender, age, expression and more. That ability allows us to react differently with a person based on the information extracted visually from his/her face. For these and other reasons, computer-based facial analysis is becoming widespread, covering applications such as identity recognition, gender determination, facial expression detection, etc.

1.4 Role of Facial Expression in Emotion Recognition

Charles Darwin was one of the first scientists to recognize that facial expression is one of the most powerful and immediate means for human beings to communicate their emotions, intentions, and opinions to each other. This was over about hundred years ago. Of course with the passage of time the human mind has expanded to such an extent that this ability to recognize and comprehend facial expressions is now passed on to the computers.

Facial expressions play a significant role in social and emotional lives. They are visually observable, conversational, and interactive signals that clarify current focus of attention and regulate interactions with the environment and other persons in vicinity. They are direct and naturally preeminent means of communicating emotions. Therefore, automated analyzers of facial expressions seem to have a natural place in various vision systems, including automated tools for behavioral research, lip reading, bimodal speech processing, videoconferencing, face/visual speech synthesis, affective computing, and perceptual man-machine interfaces. It is this wide range of principle driving applications that has lent a special impetus to the research problem of automatic facial expression analysis and produced a surge of interest in this research topic.

In the light of above discussion it is proved that facial expression plays an important role in smooth communication among individuals. The extraction and recognition of facial expression has been the topic of various researches subject to enable smooth interaction between computer and their users. In this way, computers in the future will be able to offer advice in response to the mood of the users. Computer systems with this capability have a wide range of applications in basic and applied research areas, including man-machine communication, security, law enforcement, psychiatry, education, and telecommunications.

1.5 Perception of Face Expressions

In the elaboration of expressions, the face without any expression is said to be a neutral face. It is a relaxed face without contraction of facial muscles and without facial movements. It is the state of a person's face most of the time, i.e., it is the facial appearance without any dramatic expression. In contrast, for a face with an expression, the facial muscles are somehow contracted or expanded. Hence, facial expressions are deformations of the neutral face due to a person's psychological state. These universal expressions are happiness, sadness, disgust, anger, surprise and fear and these expressions do not change too much from culture to culture.

1.6 Facial Expression Analysis

It includes both measurement of facial features and recognition of expression. The general approach to Automatic Facial Expression Analysis (AFEA) systems consists of following steps.

1.6.1 Image acquisition and preprocessing

In this process images are taken and preprocess for the enhancement. Preprocessing steps usually contains normalization, smoothing, sharpening, contrast stretching or any measures that are needed to enhance the image. Better enhancement of the image gives better localization of face and facial features.

1.6.2 Face localization and detection

As mentioned above face detection and localization is the foremost thing in the facial expression detection there are several techniques present for this task which are

- **Knowledge-based methods**. These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face localization.
- **Feature invariant approaches**. These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. The approaches involve facial features, textures, skin color and multiple features that is integration of skin color, size, and shape.
- **Template matching methods**. Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. It involves predefined face templates and deformable templates.
- **Appearance-based methods**. In contrast to template matching, the models (or templates) are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. It involves eigenface, neural networks, support vector machines, distribution based, Markov model, Naive based classifiers.

1.6.3 Feature extraction and Analysis

Expressions are basically deformation of the facial features. So this deformation information can be taken from the features that are to be extracted after face localization and detection. The features that are important and change due to any expression are position of eyebrows, eyes, wrinkles around the nose and position and shape of the lips. All these features are to extracted out either on the basis of Integral projections, landmark formation, fisher discriminant etc. Then these features are analyzed for the decision of appropriate classifier.

1.6.4 Expression recognition

Once the features are extracted out they can be given either to the neural nets or their statistical analysis generally results in the appropriate expression. General techniques for expression recognition are neural nets, SVM's, statistical methods, Bayesians Belief nets etc.

1.7 Color Spaces Used For Skin Modeling

Now for the expression recognition as one has to locate a face in the given image so this can be done by using some information about the face. There are two major types of images.

- **Gray scale images.**

  Grayscale image is simply one in which the only colors are shades of gray.In a grayscale image, each point is represented by a brightness value, ranging from 0 (black) to 255 (white), with intermediate values representing different levels of gray. In fact a `gray' color is one in which the red, green and blue components all have equal intensity in RGB space.

- **Colored images.**

  The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of gray. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color.

1.7.1 Color Spaces

A color space is a model for representing color in terms of intensity values; a color space specifies how color information is represented. It defines a one-, two-, three-, or four-dimensional space whose dimensions, or components, represent intensity values. Visually, these spaces are often represented by various solid shapes, such as cubes, cones, or polyhedron.

Colors can be defied by its attributes of brightness, hue and colorfulness. A computer may describe a color using the amounts of red, green and blue phosphor emission required to match a color. A color is thus usually specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used. A color component is also referred to as a color gamut that is the area enclosed by a color space in three dimensions. It is usual to represent the gamut of a color reproduction system graphically as the range of colors available in some device independent color space.

The few basic color spaces are

a) RGB
b) CMY(K)
c) HSV
d) YCbCr

a) RGB (Red Green Blue)

This is an additive color system based on tri-chromatic theory. Often found in systems that use a CRT to display images. RGB is easy to implement but non–linear with visual perception. It is device dependent and specification of colors is semi–intuitive. RGB is very common, being used in virtually every computer system as well as television, video etc. Figure 1-1 shows the geometry of the RGB color model for specifying colors using a Cartesian coordinate system. The grayscale spectrum, i.e. those colors made from equal amounts of each primary, lies on the line joining the black and white vertices.

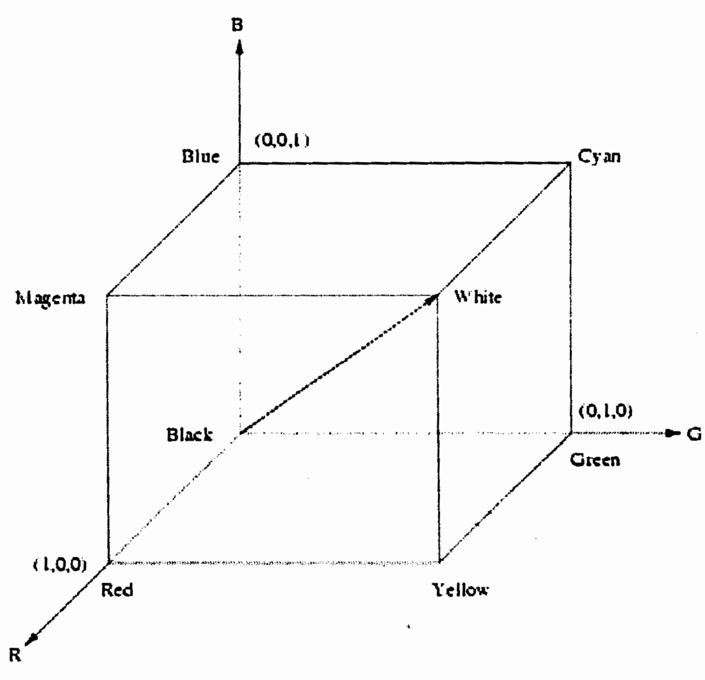


Figure 1-1The RGB color cube

b) CMY(K) (Cyan Magenta Yellow (Black))

This is a subtractive based color space and is mainly used in printing and hard copy output. The fourth, black, component is included to improve both the density range and the available color gamut. CMY(K) is fairly easy to implement but proper transfer from RGB to CMY(K) is very difficult (simple transforms are, to put it bluntly, simple).

CMY(K) is device dependent, non– linear with visual perception and reasonably unintuitive.
The relationship between the RGB and CMY models is given by [11]:

$$C = 1 - R$$
$$M = 1 - G$$
$$Y = 1 - B$$

c) HSV (Huc Saturation and Value)

This represents a wealth of similar color spaces, alternative names include HSI (intensity), HSL (Lightness), HCI (chroma / colorfulness), HVC, TSD (hue saturation and darkness) etc. Most of these color spaces are linear transforms from RGB and are therefore device dependent and non–linear. Their advantage lies in the extremely intuitive manner of specifying color. It is very easy to select a desired hue and to then modify it slightly by adjustment of its saturation and intensity. The HSI model, and the entire space of colors that may be specified in this way is shown in figure 1-2 [11].

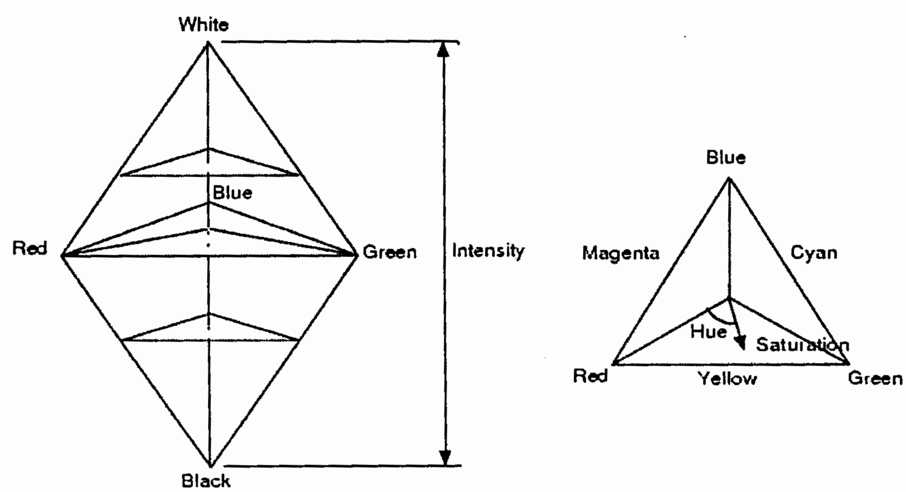


Figure 1-2 HSV color model

d) YIQ, YUV, YCbCr, YCC (Luminance - Chrominance)

These are the television transmission color spaces, sometimes known as transmission primaries. YIQ and YUV are analogue spaces for NTSC and PAL systems respectively while YCbCr is a digital standard. These colors spaces separate RGB into luminance and chrominance information and are useful in compression applications (both digital and analogue). These spaces are device dependent but are intended for use under strictly defined conditions within closed systems.

The conversion from RGB to YIQ is given by:

$$Y=0.299*R+0.587*G+0.114*B$$

$$I=0.596*R+ (-0.275*G) + (-0.321*B)$$

$$Q=0.212*R+ (-0.523*G) + (-0.311)B$$

Chapter 2

General Image Processing Techniques

2. General Image Processing Techniques

The approaches used in the project and their brief description are given.

2.1 Morphological Operations

Mathematical morphology is a set-theoretical approach to multi-dimensional digital signal or image analysis, based on shape. Fundamentally morphological image processing is very like spatial filtering. The signals are locally compared with so-called ***structuring elements*** *S* of arbitrary shape with a reference point *R*, e.g.:

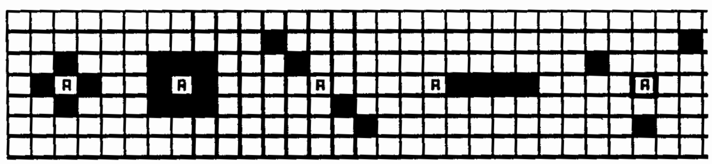


Figure 2-1: Structuring Elements

The aim is to transform the signals into simpler ones by removing irrelevant information. Morphological operations can be applied to binary and greylevel images.The morphological operations are performed by laying the structuring element on the image and sliding it across the image in a manner similar to convolution. The value of this new pixel depends on the operation performed.

Morphological Filtering uses basic operations.

1) Dilation
2) Erosion

2.1.1 Dilation

Dilation adds pixels to the boundaries of objects in an image. The number of pixels added to the objects in an image depends on the size and shape of the structuring element used to process the image. Dilation allows objects to expand, thus potentially filling in small holes and connecting disjoint objects. Dilation can repair breaks. Dilation can repair intrusions. The manner in which the dilation is calculated above presumes that a dilation can be considered to be the union of all of the translations specified by the structuring element; that is, as

$$A \oplus B = \bigcup_{b \in B} (A)_b$$

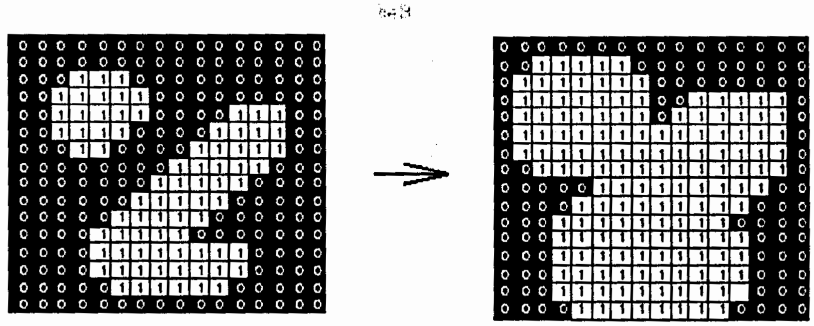


Figure2-2: Dilation using a 3 x 3 square structuring element

2.1.2 Erosion

Erosion removes pixels on object boundaries and the number of pixels removed from the objects in an image depends on the size and shape of the structuring element used to process the image. Erosion can split apart joined objects. Erosion can strip away extrusions. The manner in which the erosion is calculated above presumes that a erosion can be considered to be the intersection of all of the translations specified by the structuring element; that is, as

$$A \ominus B = \bigcap_{b \in B} (A)_b$$

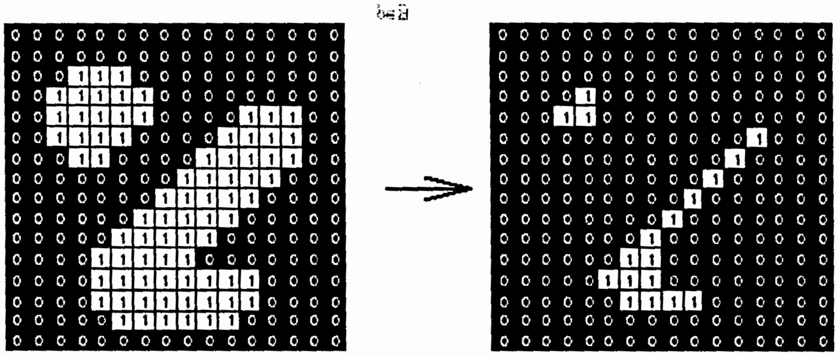


Figure 2-3: Erosion using a 3 x 3 square structuring element

More interesting morphological operations can be performed by performing combinations of erosions and dilations. The most widely used of these compound operations are:

1) Opening
2) Closing

2.1.3 Opening

Opening consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are too small to contain the structuring element. In this case the structuring element is often called a probe, because it is probing the image looking for small objects to filter out of the image. Opening of the image is given by

$$A \circ B = (A \ominus B) \oplus B$$

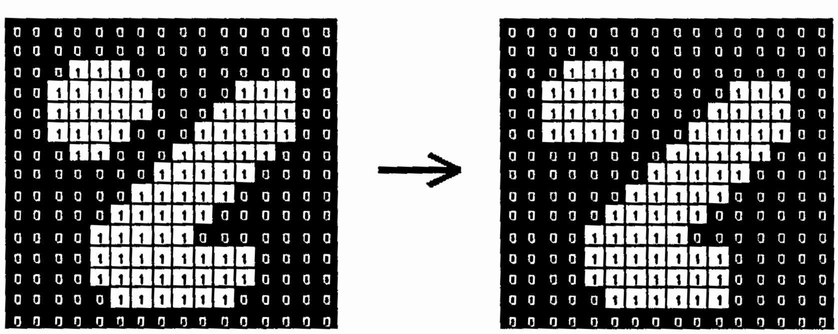


Figure 2-4: Opening using 3x3 square structuring element

2.1.4 Closing

Closing is a dilation followed by erosion (with the same structuring element). Closing also produces the smoothing of sections of contours but fuses narrow breaks, fills gaps in the contour and eliminates small holes. Closing of the image is given by

$$A \bullet B = (A \oplus B) \ominus B$$

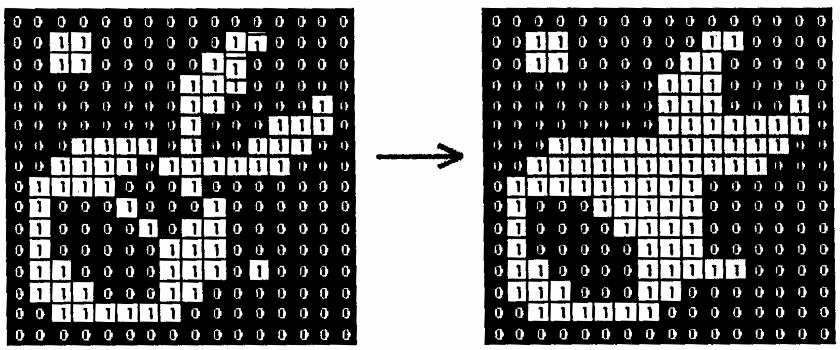


Figure 2-5: Closing using 3x3 structuring element

2.1.5 Boundary Extraction

Extracting the boundary (or outline) of an object is often extremely useful. The boundary extraction can be given simply as

$$\beta(A) = A - (A\text{-}B)$$

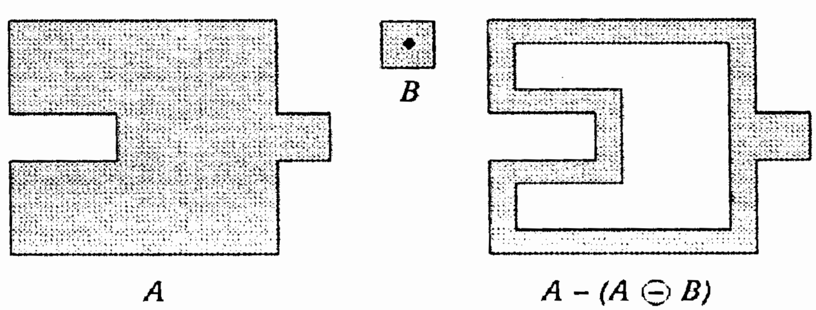


Figure 2-6: Boundary Extraction

2.1.6 Region filling

Region filling attempts to fill the background pixels enclosed with the object pixels with that of object pixels. The key equation for region filling is

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1,2,3.....$$

Where X_0 is simply the starting point inside the boundary, B is a simple structuring element and A^c is the complement of A.

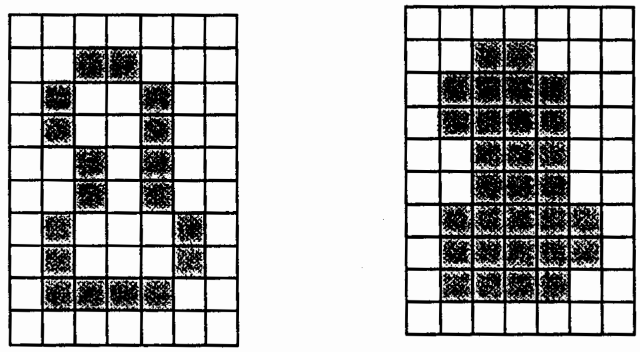


Figure 2-7: Region Filling

2.2 Filters

In image processing filters are mainly used to suppress either the high frequencies in the image, *i.e.* smoothing the image, or the low frequencies, *i.e.* enhancing or detecting edges in the image. An image can be filtered either in the frequency or in the spatial domain.

2.21 Smoothing Filters

3.2.1.1 Gaussian smoothing

The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. It also fills in the small gaps in lines, contours and planes. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where σ is the standard deviation of the distribution. It is assumed that the distribution has a mean of zero (*i.e.* it is centered about the line x=0). The distribution is illustrated as

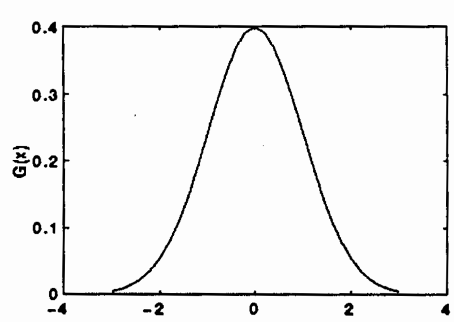


Figure 2-8: 1 D Gaussian Distribution

In two dimensions the Gaussian has the form

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This distribution is shown as

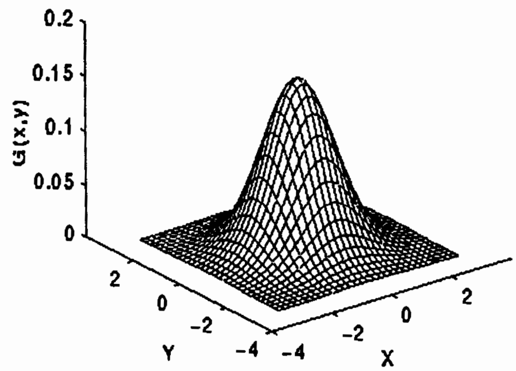


Figure 2-9: 2 D Gaussian Distribution

2.2.2 Sharpening Filters & Edge Detectors

Sharpening is used to highlight fine detail or enhance detail that has been blurred. A sharpening filter seeks to emphasize changes. Whereas edge detectors locate sharp changes in the intensity function where edges are pixels where brightness changes abruptly.

2.2.2.1 Laplacian Filter

The Laplacian is a 2D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian *L(x,y)* of an image having pixel intensity values *I(x,y)* is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

A 3 x 3 mask for 4-neighborhoods and 8-neighborhood

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2.2.2.2 Prewitt Filter

The Prewitt filter, similarly to the Sobel, Kirsch, and Robinson approximates the first derivative. The gradient is estimated in eight possible directions. The convolution result of the greatest magnitude indicates the gradient magnitude. The convolution mask of greatest magnitude indicates the gradient direction.

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

2.2.2.3 Unsharp Filter

The unsharp filter is a simple sharpening operator which derives its name from the fact that it enhances edges via a procedure which subtracts an unsharp, or smoothed, version of an image from the original image.

Unsharp masking produces an edge image $g(x,y)$ from an input image $f(x,y)$ via

$$g(x,y) = f(x,y) - f_{smooth}(x,y)$$

Where $f_{smooth}(x,y)$ is a smoothed version of $f(x,y)$.

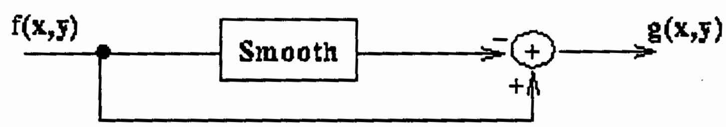


Figure 2-10: Unsharp Filter

2.2.2.4 Sobel Edge Detector

The sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. The sobel edge detector can be given as

$$N(x,y) = \sum_{k=-1}^{1} \sum_{j=-1}^{1} K(j,k)p(x-j,y-k)$$

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2.2.2.5 Canny Edge Detector

The Canny operator was designed to be an optimal edge detector. It takes as input a grey scale image, and produces as output an image showing the positions of tracked intensity discontinuities. The Canny operator works in a multi-stage process. First of all the image is smoothed by Gaussian convolution. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known *as* non-maximal suppression.

2.2.2.6 Laplacian of Gaussian

Laplacian filters are derivative filters used to find areas of rapid change (edges) in images. Since derivative filters are very sensitive to noise, it is common to smooth the image (e.g., using a Gaussian filter) before applying the Laplacian. This two-step process is called as the Laplacian of Gaussian (LoG) operation. The 2-D LoG function centered on zero and with Gaussian standard deviation has the form:

$$\mathrm{LoG}(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The LoG operator takes the second derivative of the image. Where the image is basically uniform, the LoG will give zero. Wherever a change occurs, the LoG will give a positive response on the darker side and a negative response on the lighter side. The LOG can be shown as

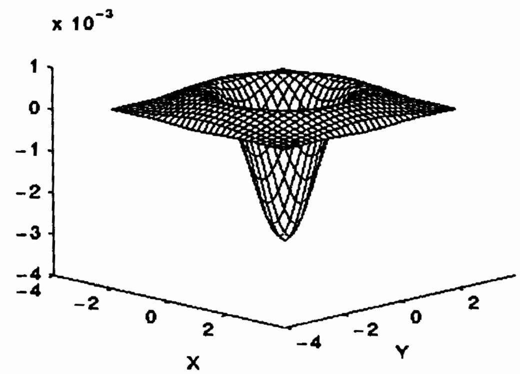


Figure 2-11: Laplacian of Gaussian curve

2.2.2.7 Gradient Operator

Gradient operators are based on local second order derivatives of the image function. The gradient operator is basically the sharpening filter applied in both horizontal and vertical direction and then taking it root mean value.

$$|grad\ g(x,y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

$$\psi = arg(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y})$$

The gradient magnitude and gradient direction are continuous image functions where arg(x,y) is the angle (in radians) from the x-axis to the point (x,y).

2.3 Integral Projections

An **integral projection** is a one-dimensional pattern, or signal, obtained through the sum of a given set of pixels along a given direction. Horizontal and vertical integral projections are most commonly used, although they can be applied on any direction.

Let *i(x,y)* be an image (greyscale or color) and *R(i)* a region in it (i.e., a set of pixels in *i(x,y)*), the **Vertical Integral Projection** of *R(i)*, denoted by $P_{VR(i)}$, is a discrete function:

$$P_{VR(i)} : \{y_{min}, ..., y_{max}\} \rightarrow \mathbf{R}$$

Defined by:

$$P_{VR(i)}(y) = \mathbf{Mean}(i(x,y)); \text{ for all } (x,y) \ \varepsilon \ R(i)$$

The **Horizontal Integral Projection** of *R(i)*, denoted by $P_{HR(i)}$, is defined in a similar way:

$$P_{HR(i)} : \{x_{min}, ..., x_{max}\} \rightarrow \mathbf{R}$$

$$P_{HR(i)}(x) = \mathbf{Mean}(i(x,y)), \text{ for all } (x,y) \ \varepsilon \ R(i)$$

Chapter 3

Literature Survey

3. Literature Survey

In the past decades there has been a considerable interest in improving all aspects of human-computer interaction (HCI). One way to achieve intelligent HCI is making computers to interact with user in the same manner as it takes place in human-human interaction. Humans naturally interact with each other through verbal (i.e. speech) and nonverbal (i.e. facial expressions, gesture, vocal tones, etc.) sign systems.

It is argued that during human-human interaction only a small part of the conveyed messages is verbally communicated, and the greatest part is nonverbally coded. Considering nonverbal communication, it is possible to say that facial expressions occupy about a half of the transmitted signals. In the context of user-friendly HCI, a face is an important source of information about the user to be analyzed by the computer. Automated analysis of a computer user's face has recently become an active research field in the computer vision community. Different vision-based schemes for intelligent HCI are currently being developed. The ability of a computer to detect, analyze and, finally, recognize a user's face has many applications in the domain of HCI. The analysis and recognition of facial expressions in the context of HCI are elements of interaction design called affective computing. The main idea of the affective computing is that the computer detects the user's affective state and takes an appropriate action, for example, offers assistance for the user or adapts to the user's needs. Proper detection of the changes in the user's facial cues is a precondition for the computer to take any emotionally or otherwise intelligent socially interactive actions towards the user. The extraction and recognition of facial expression has been the topic of various researches subject to enable smooth interaction between computer and their users. In this way, computers in the future will be able to offer advice in response to the mood of the users.

3.1 Previous work For Face Detection

Images containing faces are essential to intelligent vision-based human computer interaction, and research efforts in face processing include face recognition, face tracking, pose estimation, and expression recognition. However, many reported methods assume that the faces in an image or an image sequence have been identified and localized. To build fully automated systems that analyze the information contained in face images, robust and efficient face detection algorithms are required. Given a single image, the goal of face detection is to identify all image regions which contain a face regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is challenging because faces are nonrigid and have a high degree of variability in size, shape, color, and texture. Numerous techniques have been developed to detect faces in a single image.

3.1.1 Automatic Spatial Segmentation of Facial Regions in HSI Color Space

Arif Zuberi et al., [1] in their paper propose an automatic human facial region extraction algorithm based on color segmentation. The algorithm uses the HSI color model and has been applied on video sequences. The algorithm uses color as a visual homogeneity criterion to identify and extract the object of interest. The approach considers hue and saturation components of image for locating human skin regions and then by using edge detection techniques on hue and intensity component isolate object of interest from rest of the background. Segmentation of facial regions is achieved by using skin color to find the face. The method is based on selection of skin colored pixels in the images. Edge detection and linking is used to find the boundary of the skin colored regions and accurately segment it from the image. The largest skin cluster has to be grown, in order to encompass the entire face. The boundary of the principal skin can be traced by edge following.

3.1.2 Face Detection In Color Images

Anil K. Jain et al., [2] proposes the method detects skin regions over the entire image, and then generates face candidates based on the spatial arrangement of these skin patches. The algorithm constructs eye, mouth, and boundary maps for verifying each face candidate. They use YCbCr space for the reason to make the skin color luma independent and also enable robust detection of the dark and light skin tone colors. For the localization of facial features like eyes, mouth and face boundary etc. they use some facts of color space. An analysis of the chrominance components indicated that high *Cb* and low *Cr* values are found around the eyes, and that high *Cr* values are found in the mouth areas. Eyes usually contain both dark and bright pixels in the luma component. So the eye map is constructed using morphological operations. Eye Map in the chroma is constructed from *Cb*, the inverse of *Cr*, and the ratio *Cb/Cr*. The AND of two eye maps brightens up the eyes. The mouth region contains more red component and smaller blue component than other facial regions. Hence, the chrominance component *Cr* is greater than *Cb* near the mouth areas. We further notice that the mouth has a relatively lower response in the *Cr/Cb* feature but a high response in Cr^2. Therefore, the difference between Cr^2 and *Cr/Cb* can emphasize the mouth regions. Based on the locations of eyes/mouth candidates, our algorithm first constructs a *face boundary* map from the luma, and then utilizes a Hough transform to extract the best-fitting ellipse.

3.1.3 A Robust Skin Color Based Face Detection Algorithm

Sanjay Kr. Singh et al., [3] presents a robust skin color based face detection algorithm in which they used all the three main color spaces HSV, YCbCr and RGB for the face detection for that purpose they first compare the results in these three color spaces individually and then proposes that the best result can be obtained by combining the results of all the three color spaces.

3.2 Recent Research on Facial Expression Recognition

Computer-based recognition of facial expressions goes a long way, and various methods have been proposed.

3.2.1 Facial Expression Detection And Recognition System

W. K. Teol [4] and his companions propose the algorithm that utilizes multi-stage integral projection to extract facial features and a statistical approach to process the optical flow data to obtain the overall value for the respective feature region in the face. The optical flow computation results are processed using Kalman filtering and then the same optical flow results are processed by the statistical approach. The filtered results are given to a neural network to realize a mapping into the facial expression space. They deal with four expressions anger, sad, happy and surprise. The proposed approach uses edges extracted from the facial image that are considered as general features during feature detection process and then uses multi-stage integral projection to obtain the feature positions. Kalman filtering is applied on the optical flow values of the images sequences in an attempt to improve the overall recognition rate of the system. Kalman filter provides an efficient computational solution of the least-squares method. With a statistical approach, there is no need to track the feature movement in the subsequent image frames, as the one with the highest possibilities will indicate the average movement of the feature. Experimental results of recognition rate states that directly feeding in the Kalman filter to the neural network lead to a recognition rate of 70%. However, by applying the proposed statistical approach on the optical flow results and feeding it to the neural network lead to an improvement to 80%.

3.2.2 Automatic Detecting Neutral Face for Face Authentication and Facial Expression Analysis

Ying-li Tian and Ruud M. Bolle [5] describes an automatic system to find neutral faces in images by using location and shape features. Using these features, a window is placed in the detected and normalized face region. In that fashion, zones in this window correspond to facial regions that are loosely invariant from subject to subject. Within these zones, shape features in the form of histograms and ellipses are extracted. These features, in addition to more global distance measures, are input to a classifier to arrive at a neutral/non-neutral decision. The system has achieved an average detection rate of 97.2%. The algorithm in this paper precedes using three stages.

First phase: Face and feature detection that involves face detection and facial point feature detection. Face detection is done by the skin tone color detection and linear descriminant is then use to remove non skin pixels from gray levels. Distance from face space is then measured. Combination of the two scores is used to rank overlapping candidates that have been retained; with only the highest scoring candidate being selected and in facial point feature detection fisher discriminant and distance from feature space measurements are used to locate point features total of 6 points are detected. These point features are: pupil centers (2), eyebrow inner endpoints (2), and corners of the mouth (2).

Second phase: Location and shape features extraction. Location features are transformed into distances between the detected facial point features. These distances are relative measures, indicating relative locations, with respect to the canonical face size so the image should be normalized before this process. Then measure distances between eyebrow and eye lid, distance between upper and lower eye lid, between pupil and mouth, between corners of mouth. Divide image into three zones eyes and eyebrows region, the nose region and the mouth region. Quantize edge detection in four angular regions 0, 45, 90,135 and histograms are taken for each direction.

Third phase: Neutral face detection through neural networks. Back- propagation in the form of a three- layer neural network with one hidden layer is used to detect if a face is a neutral face or not.

3.2.3 Facial Action Recognition for Facial Expression Analysis From Static Face Images

Maja Pantic et al., [6] presents an automated system that was developed to recognize facial gestures in static, frontal- and/or profile-view color face images. A multidetector approach to facial feature localization is utilized to spatially sample the profile contour and the contours of the facial components such as the eyes and the mouth. This prototype system was aimed at automatic recognition of six basic emotions in static face images. First, static frontal and/or profile-view image of an expressionless face of the observed subject is processed. Under the assumption that input images are non-occluded, scale- and orientation-invariant face images, each subsequent image of the observed subject is processed in the following manner. The face region is extracted from the input frontal-view face image. The face-profile region is extracted from the input profile-view face image. To do so, watershed segmentation with markers is applied on the morphological gradient of the input color image. For the frontal view, the segmented face region is subjected to a multidetector processing: per facial component (eyes, eyebrows, mouth), one or more spatial samples of its contour are generated. From each spatially sampled contour of a facial component numbers of points are extracted. By performing an intra-solution consistency check, a certainty factor CF is assigned to each extracted point. On the basis of analysis of these CFs expressions are detected.

3.2.4 Automatic recognition of facial expressions using Bayesian Belief Networks

D. Datcu and L.J.M. Rothkrantz [7] in their paper addresses the aspects related to the development of an automatic probabilistic recognition system for facial expressions in video streams. The face analysis component integrates an eye tracking mechanism based on Kalman filter. The visual feature detection includes PCA oriented recognition for ranking the activity in certain facial areas. Six of the displays were based on prototypic emotions (joy, surprise, anger, fear, disgust and sadness). The important steps include eye tracking that is accomplished by using a tracking mechanism based on Kalman filter. The eye-tracking module includes some routines for detecting the position of the edge between the pupil and the iris. The process is based on the characteristic of the dark-bright pupil effect in infrared condition.

Then the next step is face representational model. The geometrical shape of each visual feature follows certain rules that aim to set the outlook to convey the appropriate emotional meaning.

In visual feature acquisition first of all, for each image a new coordinate system was set. The origin of the new coordinate system was set to the nose top of the individual. The value of a new parameter called base was computed to measure the distance between the eyes of the person in the image. All the points in the image are rotated and scaled according to the new coordinate system. Then technique used was Principal Component Analysis oriented pattern recognition for each of the three facial areas.

The final step is the inference with Baysian belief net. The data used for BBN is Static data, Dynamic data and active units. BBN sets the probability of parameter network and integrates parametric layers having different functional tasks.

3.2.5 Evaluation of Face Resolution for Expression Analysis

Ying-li Tian [8] discusses the general approach to automatic facial expression analysis (AFEA) consists of 3steps: face acquisition, facial feature extraction, and facial expression recognition. He explores the effects of different image resolutions for each step of facial expression analysis.

The face acquisition method detects both frontal and non-frontal view faces in an arbitrary scene. Face detection is evaluated on the basis of neural nets and on the set of rectangular features. Both of these approaches detect frontal view of face. The steps involved are

- Facial Feature Extraction and Representation
- Expression Recognition

Facial feature extraction and representation is done through Geometric Feature Extraction and appearance features. Geometric features present the shape and locations of facial components (including mouth, eyes, brows, nose etc.). Multistage model is used for the detection of such features.3 state model for lips, two state models for eyes and brows and cheeks have one state model. In this paper six location features are extracted for expression analysis. They are eye centers (2), eyebrow inner endpoints (2), and corners of the mouth (2). Position of eye is determine first then , the location often mouth is first predicted. Then, the vertical position of the line between the lips is found, using an integral projection of the mouth region, the horizontal borders of the line between the lips are found, using an integral projection over an edge-image. After extracting the location features, the geometric facial features can be represented by a set of parameters for expression recognition based on the line connecting the two eyes (eye-line). These parameters are the distances between the eye line and the corners of the mouth, the distances between the eye-line and the inner eyebrows, and the width of the mouth (the distance between two corners of the mouth).

Then the next step is the appearance feature extraction which is done through Gabor Wavelet Representation. Gabor wavelets are used to extract the facial appearance changes as a set of multi-scale and multi-orientation coefficients Gabor filters are applied to the difference image for the whole face. Face alignments are used to remove position noises for appearance feature extraction, the faces are normalized to a fixed distance between the center of two eyes for each face resolution. The paper investigates two type of expression recognition: FACS AUs and basic expressions. A three-layer neural network is used with one hidden layer to recognize expressions by a standard back propagation method. The inputs can be either the normalized geometric features or the appearance feature or both. The outputs are the recognized action units or six basic expressions or AUs.

3.2.6 Automatic Feature Localization in Thermal Images for Facial Expression Recognition

Leonardo Trujillo et al., [9] in their paper proposes an unsupervised Local and Global feature extraction paradigm to approach the problem of facial expression recognition in thermal images. Starting from local, low-level features computed at interest point locations, the approach combines the localization of facial features with the holistic approach. First, face localization using bi-modal thresholding is accomplished in order to localize facial features by way of novel interest point detection and clustering approach. Second, we compute representative Eigenfeatures for feature extraction. Third, facial expression classification is made with a Support Vector Machine Committee. This three step process is that

- Extraction of Facial Features
- Computing Representative Eigenfeatures
- Facial Expression Classification

Extraction of facial features involves localization of face and then localization of the facial features. First the face boundary is determined using a bi-modal thresholding technique. Then interest points are detected within the face region. An intensity based feature detector which is used in this paper extracts interest points in an image. The consideration is that any interest point operator that works directly on image intensity variations will effectively identify the image regions of interest in for that purpose Harris Operator is used. And then these detected points are grouped into clusters using k-means.

Then an anthropomorphic filter is employed to label the detected groups into left eye, right eye, mouth and face center. Eigenimage representation is used for each of the four facial regions. Eigenimages is based on Principal Component Analysis (PCA), where images are projected into a lower dimensional space that spans the significant variations (Eigenimages) among known face. The output of the combined Eigenfeature approach for image I is a composite feature set F. The idea of computing an Eigenimage representation for each region of interest, gives a Local/Global Eigenfeature representation of facial images. a different SVM is trained for each image region {left-eye, right-eye , mouth, face}. The composite feature vector F is fed to the SVM Committee, where each $\Theta_x|$ is the input to a corresponding SVM.

3.2.7 Robust Facial Expression Recognition Using Spatially Localized Geometric Model

Ashutosh Saxena et al., [10] suggested a robust facial expression recognition using spatially localized geometric model. The algorithm uses edge projection analysis for feature extraction and creates a dynamic spatio-temporal representation of the face, followed by classification through a feed-forward net with one hidden layer. A novel transform for extracting lip region for color face images based on Gaussian modeling of skin and lip color is proposed. The proposed lip transforms for colored images results in better extraction of lip region in the feature extraction stage. Thus the algorithm basically involved.

- Identification Of Feature Regions
- Feature Vector And Classification

For that purpose integral projections are used. The following generic algorithm is used for the facial feature extraction from the localized face image.

- An approximate bounding box for the feature is obtained.
- Sobel edge map is computed to obtain edges along the boundary of the feature.
- The integral projections are calculated on the edge map
- Median filtering followed by Gaussian smoothing smoothes the projection vectors so obtained.

Then first of all, eyes and eye brows are extracted and detected. Eyebrow is segmented from eye using the fact that the eye occurs below eyebrow and its edges form closed contours obtained by applying **Laplacian of Gaussian** operator at zero thresholds. These contours are filled and the resulting image containing masks of eyebrow and eye is morphologically filtered by horizontally stretched elliptic structuring elements. From the two largest filled regions, the region with higher centroid is chosen to be the mask of eyebrow. After the detection of eyes lips are detected. First of all the preprocessing is applied to enhance the lips using transform based on the probability of skin and lip color and then the generic algorithm calculates edge maps on the transformed image. Edges for lips occur both in horizontal and vertical direction. In the bounding box computed by the generic algorithm, closed contours are obtained by applying **Laplacian of Gaussian** operator at zero thresholds. These contours are filled and morphologically filtered using elliptic structuring elements to get binary mask for the lips. Nose Detection is done by approximating a bounding box for the nose lies between the eyes and the mouth. The generic algorithm uses vertical sobel edges to compute the vertical position, which is required as a reference point on face.

In the end a spatio-temporal representation of the face is created, based on geometrical relationships between features using Euclidean distance. Then this feature vector of Euclidean distances is fed into the feed forward neural network having a hidden layer.

3.2.8 Detection of Facial Landmarks from Neutral, happy, and Disgust Facial Images

Ioulia Guizatdinova and Veikko Surakka [10] suggested a method for detecting facial landmarks from neutral and expressive facial images was proposed in this paper. This paper handles only two expressions handling expressions of happiness and disgust. The algorithm involves facial land mark detection which has three major steps

Image preprocessing: In image preprocessing gray scale conversion of the image is involved and then that gray scale image is smoothed out for further processing.

Image Map Construction: The local high-contrast oriented edges were used as basic features for constructing edge maps of the image. Oriented edges were extracted by convolving the smoothed image with a set of ten convolution kernels and extracted edges are filtered. The extracted oriented edges were grouped into edge regions presumed to contain facial landmarks on the basis on neighborhood distances.

Orientation matching: The orientations of landmarks are matched on the bases of local orientation with the orientation model. It consisted of ten possible edge orientations. The following rules defined the structure of the orientation model: (a) horizontal orientations are represented by the biggest number of edge points; (b) a number of edges corresponding to each of the horizontal orientations is more than 50% bigger than a number of edges corresponding to other orientations taken separately; and (c) orientations can not be represented by zero-number of edge points.

The expressions of happiness and disgust had a marked deteriorating effect on detecting facial landmarks. It is noteworthy that the detection of nose and mouth was more affected by facial expressions than the detection of eyes. The developed method achieved a sufficiently high accuracy of 95% in detecting all four facial landmarks from the neutral images.

3.2.9 Conclusion

The literature survey on various techniques for facial expression recognitionm concludes that most of the work has done on the video streams which provide an ease as there is optical flow information giving history of the motion of feature movement but doing same task in static images is difficult. More over literature survey reveals that feature localization is majorly done on the grey scale version of the image. Some of the limitations of the papers are given below:

[4] Proceeds by extracting features by using multistage integral projection. These projections are taken on the image without localizing the face, if the face is to be localized first and then on the particular rectangle or ellipse then the projection can be

more refined. The resolution of the image and the nearer or far view of the face can effect the results of projections so this technique can be improve for better results if the image is normalized or rescale before any further processing.

[10] detects landmarks for the four face expressions , this approach emphasis to work only on the grey scale images, in this way it loses the color information in case of colored images which do provide useful information. More over landmark detection of mouth (lips) and nose is about 50% for happy and disgust expressions.

[5] works well for feature extraction but it limits to the detection of neutral and non neutral face only this research can be enhance to detect that if the face is non neutral than what basic expression it is exhibiting. Moreover no particular color space is mentioned in the paper which shows its work in RGB which is not an efficient space to be working with other color spaces can be far better than the RGB space in terms of efficiency and accuracy.

3.3 Problem Statement

A human being uses much more input information than the spoken words during a conversation with another human being: the ears to hear the words and the tone of the voice, the eyes to recognize movements of the body and facial muscles, the nose to smell where somebody has been, and the skin to recognize physical contact. In the following we will concentrate on facial expressions. Facial expressions are not only emotional states of a user but also internal states affecting his interaction with a dialogue system.

The analysis and recognition of facial expressions in the context of HCI are elements of interaction design called affective computing. The main idea of the affective computing is that the computer detects the user's affective state and takes an appropriate action, for example, offers assistance for the user or adapts to the user's needs. Proper detection of the changes in the user's facial cues is a precondition for the computer to take any emotionally or otherwise intelligent socially interactive actions towards the user.

- Facial expressions
  - ⇒ Vary in shape
  - ⇒ Highly deformable from person to person more over the environment also matters.
- Feature extraction is done in color planes of image.
- Expression detection is based on comparison of static images though it is difficult than doing it in video streams.
- Comparison of neutral face with expressive image.

Chapter 4

Proposed System

4. Proposed System

On the basis of literature survey and problem statement that is given in the previous chapter the model for the system to be developed is devised. The chapter will explore all the techniques and modules in detail that are used for system development.

The project considers six basic facial expressions excluding the neutral face. These expressions are ***happiness, sadness, disgust, anger, surprise*** and ***fear***. Universally these expressions are perceived as

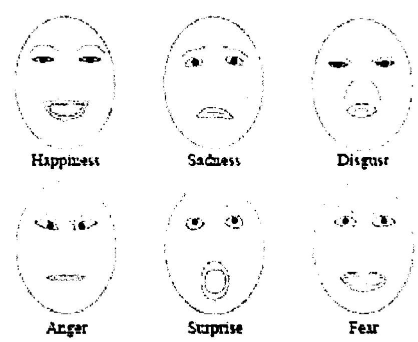


Figure 4-1: Basic Expressions [5]

It will deal with static and frontal view face images. We will work under some constraints that are background should be uniform and the background values should be different from skin. The subject should be without spectacles. He must not be mustached or beard.

4.1 Major Modules & Architectural Diagram

The modules of our project are

- Image Acquisition
- Face localization
- Features extraction
- Analysis and expression recognition

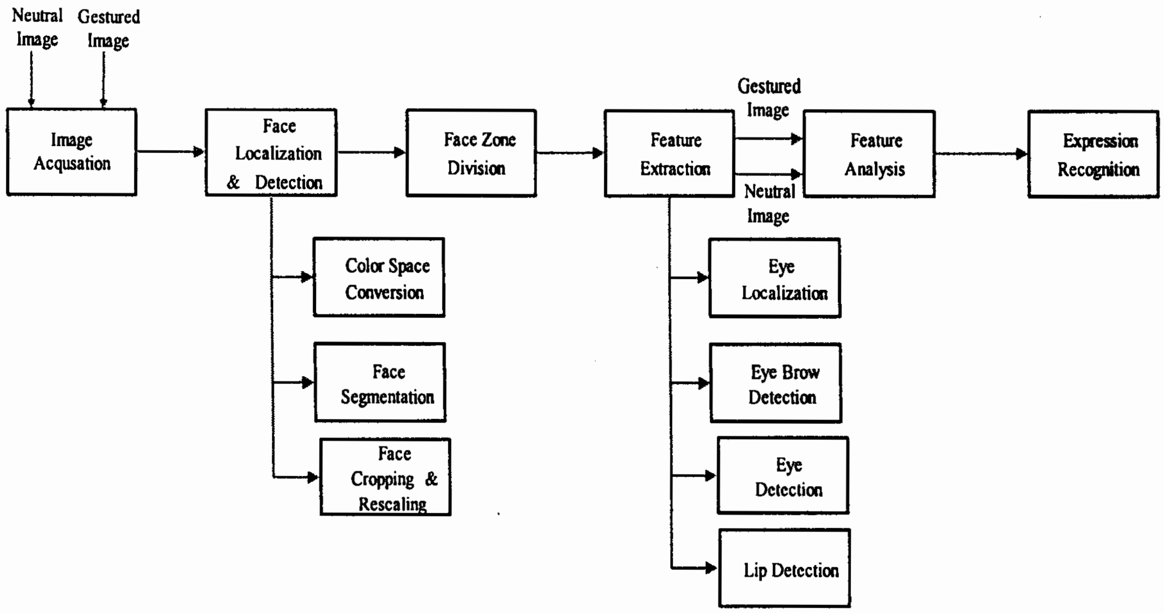


Figure 4-1: System flow diagram

The detailed description of modules is given below.

4.2 Image Acquisition

All images are taken from the digital camera. The resolution of the images is 800 X 600.

4.3 Face Localization

The localization of the face is a preliminary step, which is required to recognize the face. Given an arbitrary image, the goal of face localization is spotting whether or not a face is present, also returning its eventual location. Basic steps involves in face localization are

4.3.1 Color Space Conversion

Color has been an active area of research in image retrieval, more than in any other branch of computer vision. In fact, two aspects have a real contribution: one is that the recorded color varies considerably with the orientation of the surface, the viewpoint of the camera, the position of the illumination and which type of illumination, and the way the light interacts with the object. Second, the human perception of color has the ability to capture perceptual similarity.

The image is converted from RGB to the HSV color space.

- **Analysis of the HSV Color Space**
  A three dimensional representation of the HSV color space is a hexacone, where the central vertical axis represents the Intensity. Hue is defined as an angle in the range [0,2π] relative to the Red axis with red at angle 0, green at 2π/3, blue at 4π/3 and red again at 2π. Saturation is the depth or purity of the color and is measured as a radial distance from the central axis with value between 0 at the center to 1 at the outer surface. For S=0, as one moves higher along the Intensity axis, one goes from Black to White through various shades of gray. On the other hand, for a given Intensity and Hue, if the Saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its Hue. Saturation controls the purity of the color as the value of the saturation increases from the 0 white light is added to the color. Intensity represents particular grey level.

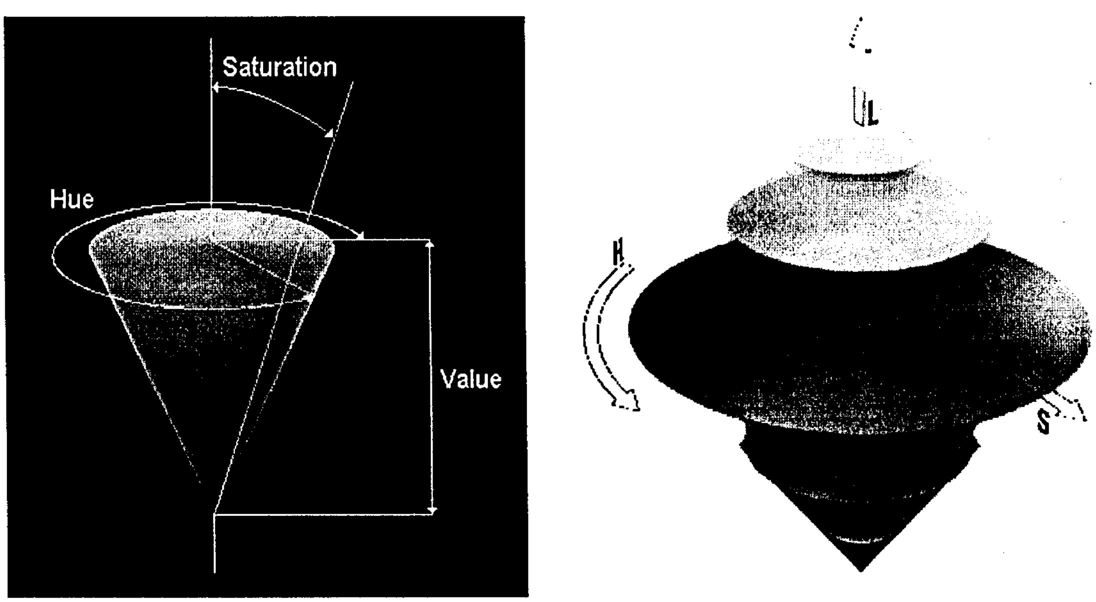


Figure 4-3: HSV color Model [18]

- **Conversion Formulae**

RGB values can be converted to HSV by using the following formulae.
Formula for hue is

$$H = \begin{cases} 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

Formula for saturation is:

$$S = \frac{MAX - MIN}{MAX}$$

Formula for value is:

$$V = MAX$$

Where MAX and MIN are the maximum and minimum values from the RGB triplet. [18]

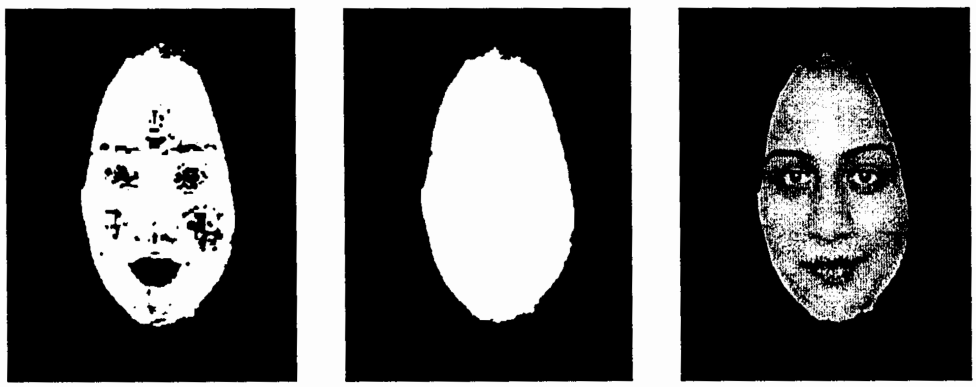
The human system can ascribe fairly constant hues to surfaces viewed in different visual contexts. The HSV representation is selected for its invariant properties. The hue is invariant under the orientation of an object with respect to the illumination and camera direction and hence more suited for object retrieval. It is highly intuitive color space from which we can extract the exact color which we perceive more over previous studies showed that the variation in skin color is due to the change in intensity values rather than hue.

(a) (b)

Figure 4-4: (a) Original Image (b) HSV of the Image

4.3.2 Face Segmentation

In the HSV color space the hue values for the skin is determined. All the things except skin are converted to background. This skin region is then on the basis of area and eccentricity is selected as the face candidate. Then morphological operations are applied which results in the proper face region.



(a) (b) (c)

Figure 4-5: (a) Segmented Face Region on the bases of HSV (b) Morphological Operations Applied (c) Segmented Face

4.3.3 Face Slant Correction

After the face segmentation face rotation is checked. Angle of rotation is determined by computing the inclination angle of the segmented region according to the centroid coordinates and the segmented region itself which is angle of major axis of the region to the x-axis.

(a) (b)

Figure 4-6: (a) Original Face (b) Rotated Face

4.3.4 Face Cropping & Rescaling

Once the face is segmented, face region is copped from the rest of the image. After cropping the face it is being rescaled for the sake of normalization. All the images are rescaled at 350 X 290.

Figure 4-7: Cropped & Rescaled Face

Algorithm

- Load the image
- Convert it into HSV space
- Scan hue values for skin region
- If pixel has skin hue value
    - Retain its value
  - Else
    - Eliminate it
- Apply smoothing on value plane to eliminate noise
- Label all the regions
- Apply morphological operations on the labeled regions
  - (dilation, fill image, opening)
- if regions passes the area and eccentricity check
    - mark it as face candidate
  - else
    - Ignore the region.
- If image is rotated/tilted
  - Correct its rotation
- Crop the face region from the whole image
- Rescale it for normalization
- Calculate its major, minor axis, centroid.
- Draw an ellipse around the face.

4.4 Face zone division

The normalized image is then divided into zones. Firstly image is divided into two zones the eye zone including eyes and the lip zone including the rest of the face. The eye zone is then further has to divide into two zones left eye zone and the right eye zone after eye localization. Major and minor axes are used for the zone division.

Figure 4-8: Face Divided into Zones

Algorithm

- Calculate left point of face
  - Left_point = xmean-minoraxis/2
- Calculate right point of face
  - Right_point=left_point+minoraxis
- Calculate top point of face
  - Top_point=ymean-majoraxis/2;
- Crop image according to the given points.

4.5 Feature Extraction

Feature extraction converts pixel data into a higher-level representation of shape. The extracted representation is used for subsequent expression categorization. It generally reduces the dimensionality of the input space. The features which we used for the expression recognition are Eyebrows, Eyes and Mouth. Eye localization is important for eyebrow and eye detection. After eye detection tip of the nose is estimated which forms equilateral triangle with the center f the eye. Once nose estimation is done the lip zone is further cropped eliminating nose region. Then comes lip localization and lip detection and its four control points are detected.

4.5.1 Eye Localization

Eye region is first of all localized for the eyebrow and eye detection. Eyes are localized on the basis of hue which does not belong to the skin region. The blobs which are not the skin region in the eye zone are then further checked for having almost same size. As the rotation is being checked for the face at the time of face localization , so the eyes should be horizontally aligned to each other and more over the distance between two eye blobs should not exceeds from the a particular limit. The blobs fulfilling all these requirements are then declared as the eyes and the eye zone is further cropped focusing only eyes. This eye zone is then divided into left and right eye zone.

Figure 4-9: Localized Eyes

(a) **(b)**

Figure 4-10: (a) Left Eye Cropped (b) Right Eye Cropped

Algorithm

- Enhance the Image
    - unsharp filtering
- Scan hue values for not skin region in face
- If pixel has skin hue value
        - Eliminate it
    - Else
        - Retain its value
- Label all the regions
- Eliminate very small object
    - Area <threshold
- Search objects having almost consistent area
- Calculate mean of both objects(fulfilling above criteria)
- Check horizontal alignment of both objects
    - If
        - xmean_obj1-xmean_obj2< 5
        - horizontally aligned

- check distance among these two objects

- o If
    - ▪ ymean_obj1-ymean_obj2< threshold
- ➢ if above conditions are true mark object as eye candidate
- ➢ Crop eye zone into left and right eye zone

4.5.2 Eyebrows Detection

Edges play an important role in human image perception and understanding Therefore, perceptually accurate segmentation of any object requires the use of edge detection [10]. We use edges detection technique for the eye brow detection. More over the fact that eye brows lie above eye also helps. Edges are found out using Laplacian of Gaussian (LoG) with zero thresholds and using standard deviation ranges 4 to 5.2[10]. The edges are then filled to form close contours. Morphological operations are then applied and the object with the less centroid is selected as the eye brow. Then initial and mid control points of the eyebrows are taken these control points are used in analysis.

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

Figure 4-11: (a) Closed Contours of Eye (b) Filled Contours (c) Detected Eyebrow (d) Detected Eye (e) Control Points of Eyebrow

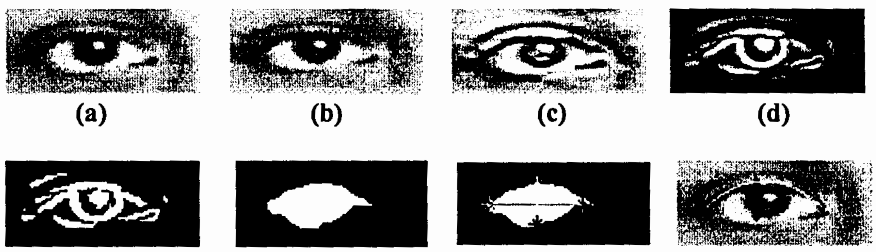
Algorithm

- ➢ Obtain edge image
    - o Using log with 0 thresholds.
    - o Standard deviation 4 ~ 5.2
- ➢ Label all regions
- ➢ Apply morphological operations on each labeled region
    - o (dilation, fill image, erosion)
- ➢ Eliminate very small object
    - o Area <threshold
- ➢ If two objects
    - ▪ select one with less centroid
    - o else
        - ▪ go to step one by increasing standard deviation
- ➢ Calculate control points
    - o For left eye brow
        - ▪ Scan image
        - ▪ Mark last pixel of the object as initial brow point
        - ▪ Calculate its centroid for middle control point.

- For right eye brow
  - Scan image
  - Mark first pixel of the object as initial brow point
  - Calculate its centroid for middle control point.

4.5.3 Eye Detection

After the eye brow detection the eye region is then further cropped; which results in the eye region. The image is then sharpened and smoothing is applied by using Gaussian filter. Double derivative of the image is taken after smoothing. Then the extended-minima transform is applied, which computes regional minima of the image. Then morphological operations are applied to obtain proper eye boundary. Once the eye boundary is detected its control point are detected by using the major and minor axis and curve is fitted on the eye according to the control points.



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

(e) (f) (g) (h)

Figure 4-12: (a) Cropped Eye (b) & (c) Enhanced Image (d) Gradient Image (e) Extended Minima (f) Morphological Operations Applied (g) & (h) Control Points Detected

Algorithm

- Crop the image
  - From the initial control point of eyebrow(eliminating eye brows)
- Smooth the image
- Calculate its double derivative(gradient)
- Calculate its regional minima
- Label all regions
- Apply morphological operations on each region
  - (dilation, thickening, bridge, fill region, erosion)
- Check object for area and solidity
  - If
    - Area <threshold
    - Eliminate object
  - If
    - Solidity <threshold
    - Apply morphological operation
- Calculate its mean, major, minor axis
- Calculate its control points
  - Left_point = xmean+minoraxis/2

- o Right_point=left_point+minoraxis
- o Top_point=ymean-majoraxis/2
- o Bottom_point=ymean+majoraxis

4.5.4 Nose Estimation

Once the eyes are being detected the nose tip is estimated by visualizing the fact that nose tip and center of both eyes form equilateral triangle. So the eye middle points are estimated, in this case we have base of the triangle. Its height is calculated by using formula

$$h = \alpha \sin 60° = \frac{1}{2}\sqrt{3}\alpha$$

Where 'α' is the length of side.[15]

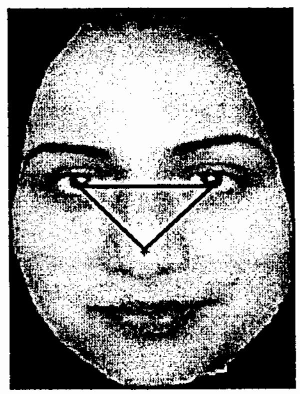


Figure 4-13: Nose estimation

Algorithm

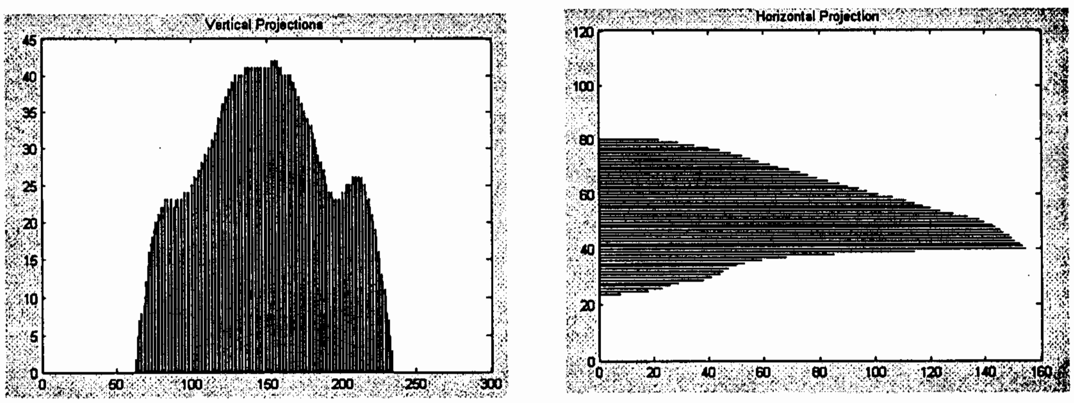
- ➢ Estimate centroid of left & right eye.
- ➢ Calculate difference between them to estimate side length.
- ➢ Calculate height of the triangle
    - o height=(sqrt(3)/2)*side_length/2;

4.5.5 Lip localization and detection

First of all Red, green and blue components are extracted, and further processing is done on the green plane. Reason for using green plane is that there are two light receptors in the human eye; these light receptors are called rods and cones. Rods are not sensitive to color and work best at low levels of illumination. The cone photoreceptors perceive the color. The cones are not evenly distributed being in the ratio of roughly 40:20:1 for RGB respectively. The green cones are most sensitive and the blue the least. As lip region changes very gradually and it is almost similar to the skin so lips are distinctive in the green plane. Lips are localized by taking integral projections of the lip zone [4]. These projections finally result in the region containing only lips. This image is then enhanced by applying image adjustment based on local statistics for sliding block operation. The local characteristic of the image used for enhancement are the global mean of image values, global variance of image, enhancement constant which is equal to 4. Finally morphological operations are applied and the maximum region is selected as the lip candidate. The top, bottom, left and control points are then detected.

(a) (b)

Figure 4-13: (a) Lower Part of the Face (b) Green Plane of Lower Part



(a) (b)

Figure 4-14: (a) Vertical Integral projection (b) Horizontal Integral Projection

(a) (b) (c)

Figure 4-15: (a) Enhanced Lip (b) Morphological Operations Applied on Lips (c) Control Points of the Lip

Algorithm

- Decompose lip zone into its RGB components
- Localize lip region
  - Vertical integral projections
  - Horizontal integral projections
- Adjust the image
  - Appling local statistics
- Label all the regions
- Apply morphological operations
- Select object on the basis of Area and Eccentricity
- Calculate its control points

- Scan image
- Mark first pixel of the object as left point
- Mark last pixel of the object as right point
- Calculate its centroid.
- Move upwards from centroid to get its top control point
- Move downwards from centroid to get its bottom control point

➢ Verify lip control points

- Bottom_cp - top_cp>threshold
- Top_cp < left_cp
- Left control point and right control point should be horizontally aligned

o If above conditions are true

- Control points are finalized

o else

- go to step 3 by applying appropriate morphological operations

4.5.6 Analysis and Expression Recognition

All the control points are obtained after lip localization. These points of the expression face are compared with the control points of the neutral face. Euclidean distances are calculated for the comparison. Aspect ratios of the eyes and lips are calculated. Mean aspect ratio of left and right eye is taken. Distance between left and right control point of lips is computed, top and bottom distance of the lip is also evaluated. Differences of these distances are taken from expression to neutral face. Left to right distance and top to bottom distances of lip are enough to differentiate happy and surprise expressions. More over aspect ratio of the eye, along with above mention distances is important to differentiate fear, anger and disgust. The aspect ratio of the lips also helps in the recognition of disgust gesture.

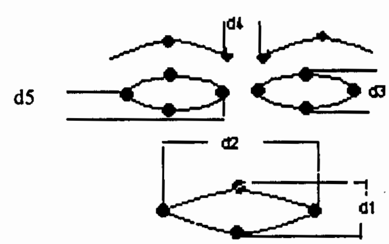


Figure 4-16 Distance Transforms

The points in the following tables are resulted from given formulae,
Euclidean distances of the following points are taken from gestured image to the neutral image.

- Eye brow initial point
- Eye brow mid point
- All four control points of eye
- All four control points of lips

All these points are marked as circle in the figure.4-16
The formula for calculating Euclidean distance is [17]

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Where 'x_2' is x-axis point of expression image and 'x_1' is x-axis point of neutral image and 'y_2' and 'y_1' are y-axis point in gestured and neutral image respectively.

Aspect ratio of eyes and lips is calculated from the following formula

Aspect ratio = width /height.
Where width is left to right distance which is d_5 in case of eye and d2 in case of lips in fig4-16, and height is top to bottom distance i.e. d3 incase of eyes and d1 in case of lips in fig 4-16.

The tables showing distances are as follows:

Happy	Upper limit	Lower limit	Difference in upper n lower lim	Mean	Standard deviation	Coefficient of variance/100
Dist b/w ini Eye Brows	12	-3	9	.69	.15	.21
Aspect ratio of eyes	0.47	0.13	0.34	0.02	0.01	0.5
Top-Bottom dist (eye) d3	2	-1	1	.07	.01	0.14
aspect ratio of lips	0.27	0.1	0.17	0.06	0.02	0.33
Left-Right dist(lip) d2	67	50	17	31.12	8.66	0.27
Top-Bottom dist(lip) d1	19	6	13	13.75	3.73	0.27

Table 4-1 Analysis of Control points for Expression of Happiness

SAD	Upper limit	Lower limit	Difference in upper n lower lim	Mean	Standard deviation	Coefficient of variance/100
Dist b/w ini Eye Brows	11	-7	18	-0.5	0.05	-0.1
Aspect ratio of eyes	0.71	0.33	0.38	0.4525	0.09	0.19
Top-Bottom dist (eye) d3	1	-2	6	-1.5	.07	-0.04
aspect ratio of lips	1.44	-0.09	1.53	0.31	0.12	0.38
Left-Right dist(lip) d2	8	-2	10	0.25	0.01	0.04
Top-Bottom dist(lip) d1	11	-2	13	-1.5	0.09	-0.06

Table 4-2 Analysis of Control points for Expression of Sadness

FEAR	Upper limit	Lower limit	Difference in upper n lower lim	Mean	Standard deviation	Coefficient of variance/100
Dist b/w ini Eye Brows	9	-5	14	5	2.13	0.42
Aspect ratio of eyes	0.58	-0.12	0.7	0.3225	0.13	0.40
Top-Bottom dist (eye) d3	1	-1	2	-1	0.04	-0.04
aspect ratio of lips	0.37	-1.03	1.4	-0.4125	0.09	-0.21
Left-Right dist(lip) d2	12	-2	14	1.75	0.58	0.18
Top-Bottom dist(lip) d1	11	0	11	6.23	1.34	0.21

Table 4-3 Analysis of Control points for Expression of Fear

Surprise	Upper limit	Lower limit	Difference in upper n lower lim	Mean	Standard deviation	Coefficient of variance/100
Dist b/w ini Eye Brows	27	8	19	6.25	3.21	0.51
Aspect ratio of eyes	1.02	1	.02	0.095	0.01	0.10
Top-Bottom dist (eye) d3	3	1	2	2.01	0.13	0.06
aspect ratio of lips	-1.03	-1.9	0.87	-1.415	0.02	-0.01
Left-Right dist(lip) d2	4	-6	10	0.75	0.01	0.01
Top-Bottom dist(lip) d1	43	23	20	31.25	6.65	0.21

Table 4-4 Analysis of Control points for Expression of Surprise

Anger	Upper limit	Lower limit	Difference in upper n lower lim	Mean	Standard deviation	Coefficient of variance/100
Dist b/w ini Eye Brows	9	-8	17	2.25	0.32	0.14
Aspect ratio of eyes	0.33	-0.04	0.37	0.1125	0.02	0.17
Top-Bottom dist (eye) d3	1	-1	2	0	0.001	NULL
aspect ratio of lips	0.6	-0.64	1.24	0.2675	0.01	0.03
Left-Right dist(lip) d2	6	-1	7	1.75	0. 18	.10
Top-Bottom dist(lip) d1	9	-6	15	0.5	0.06	0.12

Table 4-5 Analysis of Control points for Expression of Anger

Disgust	Upper limit	Lower limit	Difference in upper n lower lim	Mean	Standard deviation	Coefficient of variance/100
Dist b/w ini Eye Brows	6	-4	10	1.34	0.03	0.02
Aspect ratio of eyes	0.71	0.56	0.15	0.6425	0.061	0.09
Top-Bottom dist (eye) d3	-1	-6	5	-3.5	0.61	-0.17
aspect ratio of lips	1.06	-1.18	2.24	0.015	0.001	0.06
Left-Right dist(lip) d2	15	10	5	12.34	1.16	0.09
Top-Bottom dist(lip) d1	12	-8	20	5.25	0.50	0.09

Table 4-6 Analysis of Control points for Expression of Disgust

Algorithm for Analysis and Expression Recognition

- Compute Euclidean distance of control points.
- Calculate aspect ratio of eyes.
- Estimate height of eyes.
- Normalize aspect ratio of left and right eye.
- Compute left to right distance of lip.
- Calculate top to bottom distance of lips of lips
- Determine difference between above mention distances of expression image to neutral image.
- Classify expressions on the basis of distance distribution.

Chapter 5

Results

5. Results

Process of expression recognition involves face localization, feature (eye brows, eyes, mouth) localization and extraction. Finding control points of the features for analysis. Then analysis is performed by keeping neutral image as reference and compares relative positions of the control points in the gestured images.

5.1 Experimental Results

Some of the experimental results of the developed software are given below.

Test Face#1:

Figure 5-1 Neutral Face

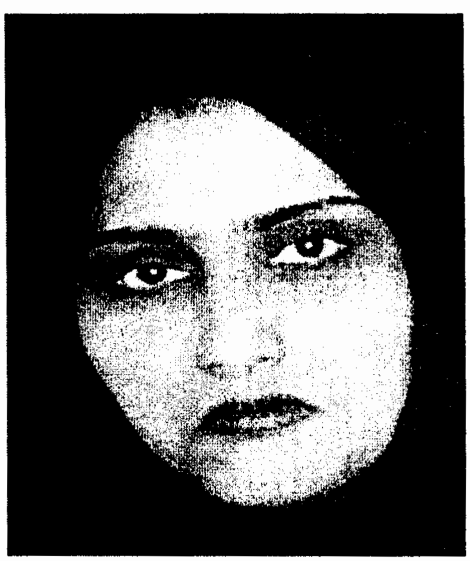


Figure 5-2 Gestured Image

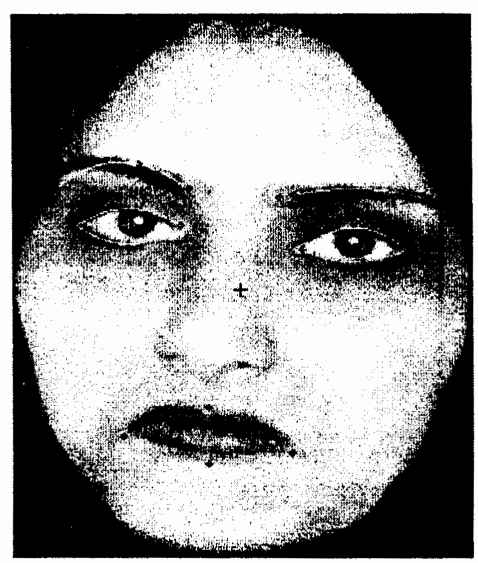


Figure 5-3 Features Extracted and Expression classified as SAD

Figure 5-4 Gestured Image

Figure 5-5 Features Extracted and Expression classified as ANGER

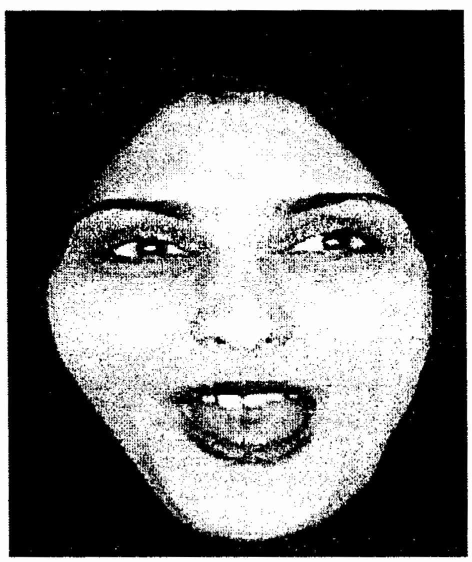


Figure 5-6 Gestured Image

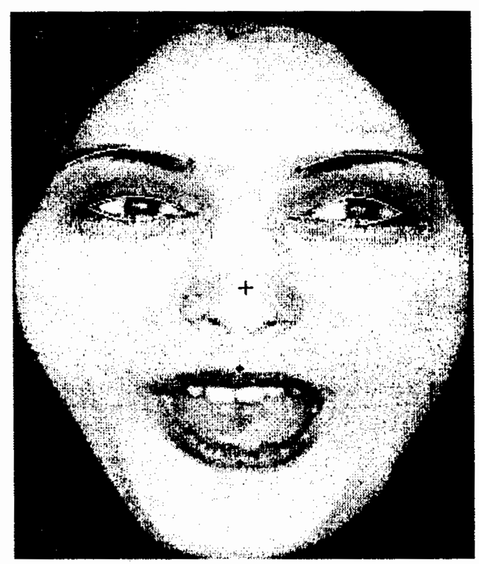


Figure 5-7 Features Extracted and Expression classified as SURPRISE

Figure 5-8 Gestured Image

Figure 5-9 Features Extracted and Expression classified as FEAR

Figure 5-10 Gestured Image

Figure 5-11 Features Extracted and Expression classified as DISGUST

Figure 5-12 Gestured Image

Figure 5-13 Features Extracted and Expression classified as HAPPY

Test Face # 2:

Figure 5-14 Neutral Image

Figure 5-15 Gestured Image

Figure 5-16 Features Extracted and Expression classified as HAPPY

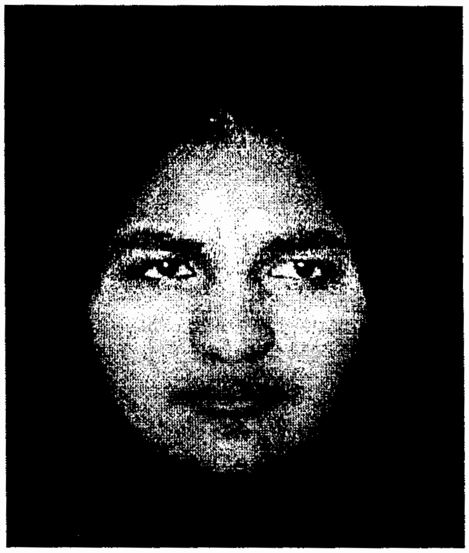


Figure 5-17 Gestured Image

Figure 5-18 Features Extracted and Expression classified as ANGER

Figure 5-19Gestured Image

Figure 5-20 Features Extracted and Expression classified as FEAR

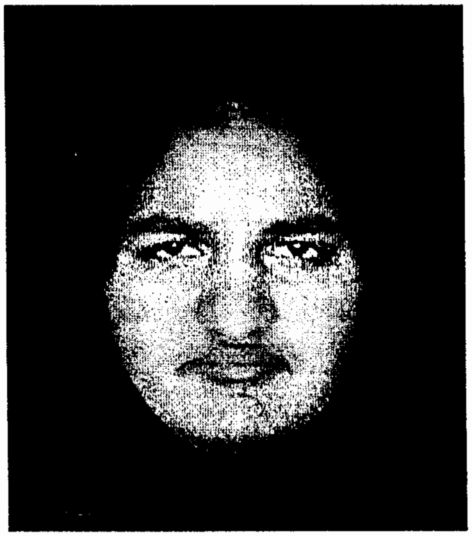


Figure 5-21 Gestured Image

Figure 5-22 Features Extracted and Expression classified as DISGUST

Figure 5-23 Gestured Image

Figure 5-24 Features Extracted and Expression classified asSurprise

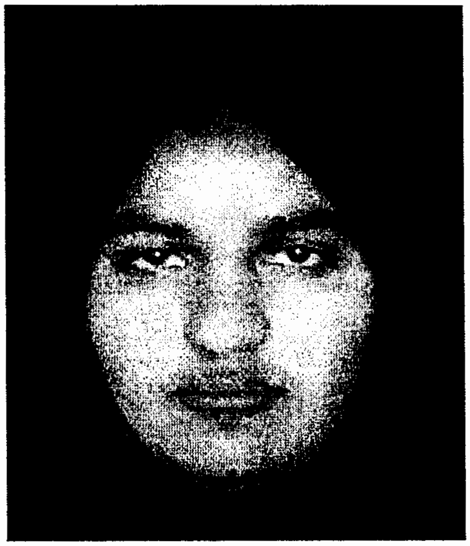


Figure 5-25 Gestured Image

Figure 5-26 Features Extracted and Expression classified as SAD

Wrong Classification:

Figure 5-27 Gestured Image

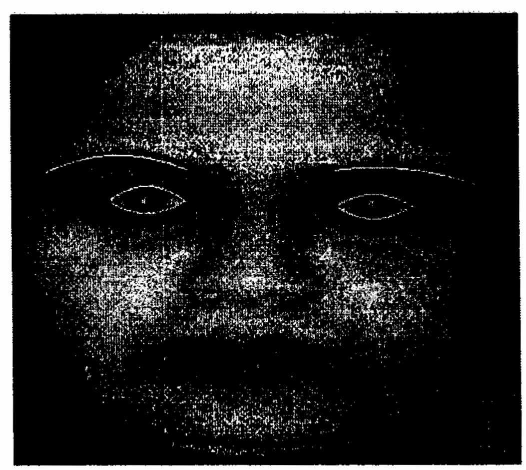


Figure 5-28 Features Extracted and Expression classified as other

5.2 Analysis of Results

We have taken 67 images of different subjects. Among these images feature localization and feature extraction is very successful. Expression classification rate is quite satisfactory. The fluctuations in the result are due to the reasons that exact expression is not demonstrated by the subject. These are in fact misleading expressions given by the subjects.

Expression/Rates	Correct Classification Rate
Happy	95~100%
Sad	78~90%
Anger	75~80%
Disgust	87~95%
Surprise	92~100%
Fear	77~90 %

Table 5-1: Classification Rate

Our results show that there are two false acceptances in case of fear, when an anger expression is recognized as fear , and one false acceptance in case of anger; when a sad expression is recognized as anger. Where as there is one case for disgust and surprise, two cases for sad and fear each, when they are not recognized as any expression because of improper expressions given by the subjects. All the happy, anger; none of these expressions is falsely accepted.

	True Acceptance	False Acceptance	False Rejection	True Rejection
Happy	13	0	54	0
Sad	11	0	56	2
Anger	9	1	57	0
Disgust	7	0	60	1
Surprise	12	0	52	1
Fear	7	1	60	2

Table 5-2: Acceptance-Rejection Rate

For further details the confusion matrix is given below. Confusion matrix in table 5-3 represents that two of the anger expression is determined as fear, because overlapping boundary of these two expressions make learning a difficult problem, one of the sad is also recognized as anger. Moreover, for statistical analysis a large data set is required for better results so the performance of the system in some cases was degraded because of limited data set. Wrong classifications are also due to the reason that expressions were not clearly portrayed by the subjects who could not act well to express the assigned expressions.

Expressions	Happy	Sad	Anger	Disgust	Surprise	Fear	other
Happy	13	0	0	0	0	0	0
Sad	0	11	1	0	0	0	2
Anger	0	0	9	0	0	2	0
Disgust	0	0	0	7	0	0	1
Surprise	0	0	0	0	12	0	1
Fear	0	0	0	0	0	7	2

Table 5-3: Confusion Matrix of Test Results

Chapter 6

Conclusion
&
Future Enhancements

6. Conclusion & Future enhancements

Facial expressions play a significant role in social and emotional lives. They are visually observable, conversational, and interactive signals that clarify current focus of attention and regulate interactions with the environment and other persons in vicinity. They are direct and naturally preeminent means of communicating emotions. Therefore, automated analyzers of facial expressions seem to have a natural place in various vision systems, including automated tools for behavioral research, lip reading, bimodal speech processing, videoconferencing, face/visual speech synthesis, affective computing, and perceptual man-machine interfaces. It is this wide range of principle driving applications that has lent a special impetus to the research problem of automatic facial expression analysis and produced a surge of interest in this research topic.

6.1 Conclusion

The project deals well with static and frontal face images. Static images are important for obtaining configurational information about facial expressions, which is essential, in turn, for inferring the expressions. Although static images are difficult to work with as in video streams optical flow information provides an ease and very diminutive work has been done on the static images. As far as frontal view face image is concerned, humans even can detect the facial expressions more accurately from the front view and also feature detection can be more accurate and easy in the frontal view.

Our research concludes the fact that better face localization leads to the better features extraction resulting in the better facial expressions recognition.

We have done purely color based feature localization (Eyes, lips). Multi detectors are used for feature extraction. Emphasis is given on the local statistics of the features for their enhancement and control point detection which gives results up to 100%.

Euclidean distances of the gestured image control points are taken with the neutral image. The statistical analysis of the feature's control points and their Euclidean distances ensuing expression results in 88% accuracy.

Comparison of our technique with the various techniques is given below:

Ying-li tian et al, [4] he had used six control points for detecting neutral or non neutral face. These 6 points are pupil centers (2), eyebrow inner endpoints (2), and corners of the mouth (2). While we are determining 14 control points which are eye brows inner points(2), eye brows middle points(2), eye's top points(2), eye's bottom points(2), eye's left and right points(4), and similarly top, bottom , left and right points of the lips(4). He had used fisher discriminents for feature extraction while we have used multi detectors for feature extraction. More over neural nets were used to determine either face is neutral or non neutral and results in 84% accuracy.

Leonardo Trujillo et al,[11] has worked on thermal images and extracted features left eye, right eye and mouth on the basis of Harris operator. No control point detection is there and had used SVM's for expression classification.

Ashutosh et al, [10] had worked on grey scale video streams while we are using colored static images. He used 6 control points which are height of eyebrow; brows distance mouth height, mouth width, upper lip curvature, lower lip curvature. Edge detection

technique is used for eye brow detection which is similar to ours but we have made different enhancements more over he had used integral projections for the localization of eye brows but we have used hue values not relating to skin for determining eyes that portion is being cropped and then edges are taken to determine eye brows. YCbCr color space is used by him for lip detection while we have localized lips in green plane. The accuracy in his algorithm is 94%.
The comparison of these researches with ours can be viewed in tabular form in table 6-1, 6-2, 6-3 respectively.

Ying Li tian et al, [4]	AFER
Used both colored and grey scale images	Colored images only
Fisher discriminants for feature extraction	Multi detectors, edge detection (eye brows), gradient and regional minima(eyes), integral projection and local statistics (lips)
6 control points which are pupil centers (2), eyebrow inner endpoints (2), and corners of the mouth (2)	14 control points which are eye brows inner points(2), eye brows middle points(2), eye's top points(2), eye's bottom points(2), eyes left and right points(4), and similarly top, bottom , left and right points of the lips(4).
Classify neutral and non neutral face	Classify into six universal expressions happy, sad, disgust, fear, anger and surprise in comparison with neutral face.
Neural nets	Statistical analysis of difference in neutral and gestured image control points
84% results	88%results

Table 6-1 comparison with [4]

Leonardo Trujillo et al [11]	AFER
Uses thermal images	Colored images only
Harris operator is used for feature extraction and extracted features are mouth, left and right eye.	Multi detectors, edge detection (eye brows), gradient and regional minima(eyes), integral projection and local statistics (lips)
No control point detection	14 control points are detected
SVM's and PCA are used for expression classification	Statistical analysis of difference in neutral and gestured image control points

Table 6-2 comparison with [11]

Ashutosh et al [4]	AFER
Used grey scale images	Colored images only
Used videostreams	Static images are used
Integral projections for feature extraction	Features are localized on the basis of color components. Eye(hue) lips(green plane)
Edge detection for eye brows detection	Edge detection [11].
YCbCr space used for lip detection	Green plane is used for lip detection
6 control points which are height of eyebrow; brows distance mouth height, mouth width, upper lip curvature, lower lip curvature	14 control points which are eye brows inner points(2), eye brows middle points(2), eye's top points(2), eye's bottom points(2), eyes left and right points(4), and similarly top, bottom , left and right points of the lips(4).
Neural nets	Statistical analysis of difference in neutral and gestured image control points
90% results	88%results

Table 6-3 comparison with [10]

We have worked with some constraints that image should be single faced, frontal view and subject should not have moustaches and beard. More over background should be plane and preferably be of dark shade. We have done all the work on the local images except a few images taken from net. The software will work perfectly with the Asian people but it may have some exception with other continents.

6.2 Future Enhancements

The technique can be enhanced to multi face images. Furthermore work can be done on the subjects having beard, moustaches, or wearing glasses with complex background. More over real time applications can also be made. Cropping of features after localization may be enhanced by some other method. Neural nets or some other technique such as SVM's or HMM's can be used for the analysis of the control point. Development of same project in some other tool such as VC++ or Dot net frame work can improve the efficiency by reducing its time complexity.

Appendix A

Implementation

A. Implementation

Implementation includes all the details that were required to make the system operational. The development tools and technologies to implement the system and also reasons for selecting particular tool are discussed. Then the modules being translated into the implementation tool will be descried.

Software selection is very important step in developing a computer based system. The software that is used is capable of meeting the requirement of the proposed system. After considering the number of tools available these days such as Visual Basic, Visual C++.Net, Visual C, Java we choose MATLAB.

A-1 Face localization

This module accepts colored 'jpg' image as input and returns cropped and normalized image. Its sub modules are

A-1.1 Conversion to HSV

This function takes three dimensional matrix containing Red, Green, and Blue planes as input and return three planes in form of Hue, saturation and value. These three planes are then decomposed into independent components.

```
im=rgb2hsv(x);
h=im(:,:,1);
s=im(:,:,2);
v=im(:,:,3);
```

A-1.2 Segmentation Of Skin And Non Skin Region

This function takes all three planes, Hue saturation and value as input. Its fourth element describes whether to retain skin color values in case of face localization or non skin values in case of eye localization. After scanning skin or non skin models it then recombine all three planes and convert it again into its RGB format.

```
function [im_fhsv,im_frgb]=seg_skin(h,s1,v,val)

if  nargin < 4
    warning ('val is not determined...'); val=1;
end
[x1,y]=size(h);
if val==1
    for z=1:x1
        for j=1:y
         if((h(z,j)>=0.02)&(h(z,j)<=0.1)))
                h1(z,j)= h(z,j);
                s2(z,j)= s1(z,j);
                v1(z,j)= v(z,j);
                h2(z,j)= 1;s22(z,j)=1;v2(z,j)= 1;
            else
                h1(z,j)=0; s2(z,j)=0; v1(z,j)=0;
```

```
                    h2(z,j)=0;s22(z,j)=0;v2(z,j)=0;
               end
            end
        end
    end
    if val==2
        for z=1:x1
            for j=1:y
                if  (~((h(z,j)>=0.02)&(h(z,j)<=0.07)))&(h(z,j)~=0)
                    h1(z,j)= h(z,j);
                    s2(z,j)= s1(z,j);
                    v1(z,j)= v(z,j);
                    h2(z,j)= 1;s22(z,j)= 1;v2(z,j)= 1;
                else
                    h1(z,j)=0;s2(z,j)=0;v1(z,j)=0;
                    h2(z,j)= 0;s22(z,j)= 0;v2(z,j)= 0;
                end
            end
        end
    end
    fil = fspecial('gaussian',[7 7]);
    v1=imfilter(v1,fil);
    % shows the segmented image
    f1=cat(3,h1,s2,v1);
    f12=hsv2rgb(f1);
    f2=cat(3,h2,s22,v2);
    f22=hsv2rgb(f2);
```

A-1.3 Label Regions

This function takes segmented image as input, convert it into bi-level image, fill holes and then label each object in the image so that they can be manipulated differently.

```
function [L,num]=label_reg(im_l)

s=graythresh(im_l);
BW = im2bw(im_l, s);
% fill holes
BW5 = imfill(BW,'holes');
[L,num] = bwlabel(BW5,8);
RGB = label2rgb(L);
RGB2 = label2rgb(L, 'spring', 'c', 'shuffle');
figure,imshow(RGB2);
```

A-1.4 Face Detection

This function accepts all the labeled images along with total number of labels apply morphological operations on the labeled images, scan each of them for particular area and eccentricity to select face candidate. If more than one object passes the test then one having larger area is selected.

```
function [faces,xmf,ymf]=face_seg(L,num)

s = struct('im', {});
for z=1:num
    [r1,c1]= find(L==z);
```

```
        BW2 = bwselect(L,c1,r1,8);
        BW2=bwmorph(BW2,'dilate');
        obj(z) = regionprops(double(BW2),'all');
        ar(z) = obj(z).Area;
        t(z)=obj(z).Eccentricity;
    if((ar(z)>=900)&(obj(z).Eccentricity>.558)&(obj(z).Eccentricity<=.
    852))
            ecc=obj(z).Eccentricity;
            orient12= obj(z).Orientation;
            BW21 = bwselect(L,c1,r1,8);
            BW23=BW21;
            s(count).im=BW23;
            arw(count)=obj(z).Area;
            orient(count)=orient12;
            ore2=orient12;
            count=count+1;
        end
    end
    if count>2
        maxa= max(max(arw));
        ind=find(arw==maxa);
        BW23=s(ind).im;
        ore2=orient(ind);
    end
    [xmean, ymean] = center(BW23);
    figure, imshow(BW23);
    cBW=bwmorph(BW23,'dilate',2);
    cBW=imfill(cBW,'holes');
```

A-1.5Finding Center of Face

This function takes bi-level image as input and return centre of the image in form of its x-axis and y-axis.

```
function [xmean, ymean] = center(bw)

bw=bwfill(bw,'holes');
area = bwarea(bw);
[m n] =size(bw);
bw=double(bw);
xmean =0; ymean = 0;
for i=1:m,
    for j=1:n,
        xmean = xmean + j*bw(i,j);
        ymean = ymean + i*bw(i,j);
    end;
end;
xmean = xmean/area;
ymean = ymean/area;
xmean = round(xmean);
ymean = round(ymean);
```

A-1.6 Estimating Orientation of Face

This function finds rotation angle of the object by taking center of the object and the object itself as an input.

```
function [theta] = orient(bw,xmean,ymean)

bw=double(bw);
for i=1:m,
    for j=1:n,
        a = a + (j - xmean)^2 * bw(i,j);
        b = b + (j - xmean) * (i - ymean) * bw(i,j);
        c = c + (i - ymean)^2 * bw(i,j);
    end;
end;
b = 2 * b;
theta = atan(b/(a-c))/2;
% change from radians to degrees.
theta = theta*(180/pi);
```

A-1.7 Face Rotation

```
function [bw_r,angle,xmf,ymf]=face_rotation(BW22,xmean,ymean)

angle = orient(BW22,xmean,ymean)
if (angle>0) angle=-1*angle; end;
bw_r = imrotate(BW22, angle, 'bilinear');
obj=regionprops(double(bw_r),'all');;
[xmf,ymf]=center(bw_r);
orient_flip=obj.Orientation;
angle_flip=orient(bw_r,xmf,ymf);
```

A-1.8 Calculating Object Properties

This module accepts bi-level image as input and returns its center, major, minor axis and orientation.

```
function [xmr,ymr,mjr,mnr,orl]=face_loc(bw_r)

bw_r=im2bw(bw_r);
[xmean1, ymean1] = center(bw_r);
[L1,num1] = bwlabel(bw_r,8);
[rf,cf]= find(L1==1);
bv = bwselect(L1,cf,rf,8);
obj1 = regionprops(double(bv),'all');
mjaxis1=obj1.MajorAxisLength;
mnaxis1=obj1.MinorAxisLength;
or1=obj1.Orientation;
convA_face=obj1.ConvexArea;
Sol_face=obj1.Solidity;
```

A-1.9 Restoring Skin Values:

This function take a bi-level image, hue, saturation and value plane along with angle and return the image resorting skin values of the object.

```
function [rgb2]=face_locl(BW22,h,sl,v,xl,y,angle)

for z=1:xl
    for j=1:y
```

```
                if  ((BW22(z,j)==1))
                    h3(z,j)= h(z,j);
                    s3(z,j)= s1(z,j);
                    v3(z,j)= v(z,j);
                else
                    h3(z,j)=0; s3(z,j)=0;v3(z,j)=0;
                end
            end
          end
        hsv3=cat(3,h3,s3,v3);
        h3r = imrotate(h3, angle, 'bilinear');
        s3r= imrotate(s3, angle, 'bilinear');
        v3r= imrotate(v3, angle, 'bilinear');
        hsv3r=cat(3,h3r,s3r,v3r);
        rgb2=hsv2rgb(hsv3r);
```

A-1.10 Bounding Rectangle

This function takes a black and white object as input and returns the top, bottom, right and left coordinates enclosing rectangle around the object.

```
function [left, right, up, down] = recsize(A)

[m n] = size(A);
left = -1;
right = -1;
up = -1;
down = -1;
for j=1:n,
    for i=1:m,
        if (A(i,j) ~= 0)
            left = j;
            break;
        end;
    end;
    if (left ~= -1) break; end;
end;
for j=n:-1:1,
    for i=1:m,
        if (A(i,j) ~= 0)
            right = j; break; end;
    end;
    if (right ~= -1) break; end;
end;
for i=1:m,
    for j=1:n,
        if (A(i,j) ~= 0) up = i; break; end;
    end;
    if (up ~= -1) break; end;
end;
for i= m:-1:1,
    for j=1:n,
        if (A(i,j) ~= 0) down = i; break; end;
    end;
    if (down ~= -1) break; end;
end;
if (left == -1) left =1; end;
```

```
if (right == -1) right = n; end;
if (up == -1) up=1; end;
if (down == -1) down = m; end;
```

A-1.11 Image Rescaling

This function take image and new size in which it has to rescale as input and returns rescaled image along with check value that it is rescaled well.

```
function [g,check] = imgrescale(data, newsize)
try
    factor = (oldsize(1:2)-1)./(newsize-1);
    u = 0:newsize(1)-1;
    v = 0:newsize(2)-1;
    [U, V] = ndgrid(u, v);
    % transform new grid vectors to old grid size
    u = u.*factor(1) + 1;
    v = v.*factor(2) + 1;
    U = U.*factor(1); U = U - fix(U);
    V = V.*factor(2); V = V - fix(V);
     if ~RGB
        g = (V-1).*((U-1).*data(floor(u), floor(v)) - ...
            U.*data(ceil(u), floor(v))) - ...
            V.*((U-1).*data(floor(u), ceil(v)) - ...
            U.*data(ceil(u), ceil(v)));
    else
        U = repmat(U, [1 1 3]);
        V = repmat(V, [1 1 3]);
        g = (V-1).*((U-1).*data(floor(u), floor(v), :) - ...
            U.*data(ceil(u), floor(v), :)) - ...
            V.*((U-1).*data(floor(u), ceil(v), :) - ...
            U.*data(ceil(u), ceil(v), :));
    end
catch
    errmsg = lasterr;
    if(strfind(errmsg, 'Index exceeds matrix dimensions'))
        check=1
        g=1;
    end
end
```

A-1.12 Plotting Ellipse

```
function
handle=plot_ellipse(major,minor,inc,phase,pos,options,type)

if nargin < 7    type=1; end;
if nargin < 6    options='b'; end;
if nargin < 5    pos=[0 0]; end;
if major < minor
    warning('major axis cant be less than minor...')
end
t=linspace(0,2*pi,100);
x=major*cos(t-phase*pi/180);
y=minor*sin(t-phase*pi/180);
[th,r]=cart2pol(x,y);
```

```
[x,y]=pol2cart(th+inc*pi/180,r);
if type == 1 | type == 2    a=major/10; end;
if type == 3
    a=major/20; % length
end
Fi=30;      % angle.
teta=atan2(y(end)-y(end-1),x(end)-x(end-1))*180/pi;
fi=(180-Fi+teta)*pi/180;
fii=(180+Fi+teta)*pi/180;
if type == 3
    aux=3; % extra arrow point
    teta_aux=atan2(y(aux)-y(aux-1),x(aux)-x(aux-1))*180/pi;
    fi_aux=(180-Fi+teta_aux)*pi/180;
    fii_aux=(180+Fi+teta_aux)*pi/180;

    aux2=6; % extra arrow point
    teta_aux2=atan2(y(aux2)-y(aux2-1),x(aux2)-x(aux2-1))*180/pi;
    fi_aux2=(180-Fi+teta_aux2)*pi/180;
    fii_aux2=(180+Fi+teta_aux2)*pi/180;
end
Fi=Fi*pi/180;
P1=[x(1) y(1)];
P2=[P1(1)+a*cos(fi)/cos(Fi)  P1(2)+a*sin(fi)/cos(Fi)  P1(end)];
P3=[P1(1)+a*cos(fii)/cos(Fi) P1(2)+a*sin(fii)/cos(Fi) P1(end)];
if type == 3
    P1_aux=[x(aux) y(aux)];
    P2_aux=[P1_aux(1)+a*cos(fi_aux)/cos(Fi)
P1_aux(2)+a*sin(fi_aux)/cos(Fi)  P1(end)];
    P3_aux=[P1_aux(1)+a*cos(fii_aux)/cos(Fi)
P1_aux(2)+a*sin(fii_aux)/cos(Fi) P1(end)];
    P1_aux2=[x(aux2) y(aux2)];
    P2_aux2=[P1_aux2(1)+a*cos(fi_aux2)/cos(Fi)
P1_aux2(2)+a*sin(fi_aux2)/cos(Fi)  P1(end)];
    P3_aux2=[P1_aux2(1)+a*cos(fii_aux2)/cos(Fi)
P1_aux2(2)+a*sin(fii_aux2)/cos(Fi) P1(end)];
end
if type == 1
    x=pos(1)+[0 x NaN P2(1) P1(1) P3(1)];
    y=pos(2)+[0 y NaN P2(2) P1(2) P3(2)];
elseif type == 2
    x=pos(1)+[x NaN P2(1) P1(1) P3(1)];
    y=pos(2)+[y NaN P2(2) P1(2) P3(2)];
elseif type == 3
    x=pos(1)+x;
    y=pos(2)+y;
    x1=x(1);
    y1=y(1);
    x_aux=pos(1)+[P2_aux(1) P1_aux(1) P3_aux(1)];
    y_aux=pos(2)+[P2_aux(2) P1_aux(2) P3_aux(2)];
    x_aux2=pos(1)+[P2_aux2(1) P1_aux2(1) P3_aux2(1)];
    y_aux2=pos(2)+[P2_aux2(2) P1_aux2(2) P3_aux2(2)];
    x=[x NaN x_aux nan x_aux2];
    y=[y NaN y_aux nan y_aux2];
end
h=ishold;
if length(pos) == 2
    if type == 3
```

```
            handle.poin1=plot(x1,y1,'k.')
        else
            handle=plot(x,y,options);
        end
    elseif length(pos) == 3
        if type == 3
            handle.line=plot3(x,y,pos(3)*ones(size(x)),options)
            hold on
            handle.poin1=plot3(x1,y1,pos(3),'k.')
        else
            handle=plot3(x,y,pos(3)*ones(size(x)),options)
        end
    else
        warning('wrong pos dimensions...')
    end
    if h~=1
        hold off
    end
```

A-2 Face Zone Division

This module divides face into two zones, eye zone and lip zone on the basis of its major and minor axis.

```
function imag=face_zones(rgb2,xmean1,ymean1,mnaxis1,mjaxis1,val,offset)

    if nargin<7   offset=0; end;
    p1=xmean1-mnaxis1/2;
    p2=p1+mnaxis1;
    plot(p1:p2,ymean1,'g');
    hold on;
    l1=ymean1-mjaxis1/2;
    l2=l1+mjaxis1/2;
    plot(xmean1,l1:l2,'g');
    im_eye=imcrop(rgb2 ,[0 0 u ymean1]);
    [ue,ve]=size(im_eye);
    im_el=imcrop(im_eye ,[0 0 xmean1 ve]);
    im_er=imcrop(im_eye ,[xmean1 0 ue ve]);
    im_lips=imcrop(rgb2 ,[0 offset+10 u v-offset-10]);
    if (val==1)
        imag=im_eye;
    else if(val==2)
            imag=im_lips;
        end
    end
```

A-3 Eye Localization

This module accepts eye zone as the input and localizes eyes on the basis of not skin region code given in 5.2.2.and the selection of eye candidates depend on the horizontal alignment and distance between them.

A-3.1 Selection of Eye Candidates

This function takes number of labels and all the labels as input. Among all of them it selects two as eye candidates which have consistent area, horizontally aligned and distance between them is within threshold.

```
function [im1,im2]=can_eye(L,num,rgb2)

s = struct('im', {});
r1=0;c1=0;count=1;index=1;index1=1;cen=0;
for z=1:num
    cn=0;
    [r1,c1]= find(L==z);
    BW2 = bwselect(L,c1,r1,8);
    area2(count)=bwarea(BW2);
    if(bwarea(BW2)>200)
        area1(count)=bwarea(BW2)
        ele(count)=z
        figure,imshow(BW2),title(z);
        count=count+1;
    end
end
if(count<=2) ele=0; area1=0; count=1;
    for z=1:num
        [r1,c1]= find(L==z);
        BW2 = bwselect(L,c1,r1,8);
        area1(count)=bwarea(BW2)
        ele(count)=z
        figure,imshow(BW2),title(z);
        count=count+1;
    end
end
for z=1:b
    for z1=z+1:b
        if ((area1(z)>=area1(z1)-700)&(area1(z)<=area1(z1)+700 ))
            area1(z)
            area1(z1)
            [r1,c1]= find(L==ele(z));
            BW = bwselect(L,c1,r1,8);
            [r2,c2]= find(L==ele(z1));
            BW2 = bwselect(L,c2,r2,8);
            [x1,y1]=center(BW);
            [x2,y2]=center(BW2);
            dist=x2-x1
            dist2=y2-y1
          if(((dist>=95)&(dist<=150)) &((dist2>=-8)&(dist2<=8)))
                xmean1=x1;
                ymean1=y1;
                xmean2=x2;
                ymean2=y2;
                s(index).im=BW;
                indnum(index)=z;
                cen(index)=y1;
                index=index+1;
                indnum(index)=z1;
                s(index).im=BW2;
                cen(index)=y2;
```

```
                    index=index+1;
                    disr(index1)=dist;
                    index1=index1+1;
                end
            end
        end
    end

    if (index1>2)
        mind=min(min(disr));
        ind=find(disr==mind);
        maxc=max(max(cen));
        indc=find(cen==maxc);
        if(indnum(1)==indnum(3)|indnum(2)==indnum(4))
            if (ind==1)
                im1=s(1).im;
                im2=s(2).im;
            else
                im1=s(3).im;
                im2=s(4).im;
            end
        else
            if (indc==1 | indc==2)
                im1=s(1).im;
                im2=s(2).im;
            else
                im1=s(3).im;
                im2=s(4).im;
            end
        end
    else if (index1<2)
            for z=1:b
                for z1=z+1:b
          if ((areal(z)>=areal(z1)-200)&(areal(z)<=areal(z1)+1200 ))
                        areal(z);
                        areal(z1);
                        [r1,c1]= find(L==ele(z));
                        BW = bwselect(L,c1,r1,8);
                        [r2,c2]= find(L==ele(z1));
                        BW2 = bwselect(L,c2,r2,8);
                        [x1,y1]=center(BW);
                        [x2,y2]=center(BW2);
                        dist=x2-x1;
                        dist2=y2-y1;
           if(((dist>=90)&(dist<=150)) &((dist2>=-20)&(dist2<=20)))
                            xmean1=x1;
                            ymean1=y1;
                            xmean2=x2;
                            ymean2=y2;
                            im1=BW;
                            index=index+1;
                            im2=BW2;
                            index=index+1;
                            disr(index1)=dist;
                            index1=index1+1;
                        end
                    end
```

```
                end
            end
        else
            im1=s(1).im;
            im2=s(2).im;
        end; end;
    if(index1<2) [im1,im2]=can_eye12(L,num,rgb2); end;
```

A-4 Eye Brow Detection

This module locates eye brows using close contours and extracts its control points.

A-4.1 Finding Close Contours

Edge image is obtained by using log filter with zero threshold and using standard deviation of 4. These contours are then filled.

```
im_el_g = edge(imel,'log',0,4);
figure,imshow(im_el_g),title('log of e l');
im_el_g =imfill(im_el_g ,'holes');
figure,imshow(im_el_g);
[Lel,numel]=label_reglips(im_el_g);
```

A-4.2 Selection of Eye Brow Candidate

The remaining regions are then labeled and label image, number of labels, and eye image as given input to the function, among which undesired objects were removed on the basis of area and among remaining two the one with less centroid is selected and returned along with its middle x and y point. Check will determine the success full return of the function.

```
function [fill_e,xmb,ymb,check]=can_brow(L,num,ime)

s = struct('im', {});
rw = struct('rows', {});
for z=1:num
    [r1,c1]= find(L==z);
    BW21 = bwselect(L,c1,r1,8);
    im_elb=brow_enh(BW21);
    if bwarea(im_elb)==0  continue; end;
    [Lel,numel]=label_reglips(im_elb);
    for z1=1:numel
        [r11,c11]= find(Lel==z1);
        BW2 = bwselect(Lel,c11,r11,8);
        area1(cn)=bwarea(BW2);
        if(area1(cn)>550)
            [xmean, ymean] = center(BW2);
            obj=regionprops(double(BW2),'all');
            obj.Eccentricity;
            x1(count)=xmean;
            y1(count)=ymean;
            in(count)=z1;
            in1(count)=z;
            rw(count).row=r11;
            s(count).im=BW2;
```

```
                count=count+1
                ce(1,count)={BW2};
            end;
            cn=cn+1;
        end;
    end;
    if(count<3) check=1; fill_e=1;xmb=1;ymb=1;
    else
        if(count>3)
            check=2;
            fill_e=1;xmb=1;ymb=1;
            warning('more objects');
        else
            if (y1(2)>y1(1))
                ime_b = s(1).im;
                if(max(rw(1).row)>=y1(2))
                    check=1;
                    fill_e=1;xmb=1;ymb=1;
                else
                    imeb=ime_b;
                    [xmb,ymb]=center(imeb);
                    brg_e=bwmorph(imeb,'bridge',1);
                    fill_e=imfill(brg_e,'holes');
                end
            else
                ime_b =s(2).im;
                if(max(rw(2).row)>=y1(1))
                    check=1;
                    fill_e=1;xmb=1;ymb=1;
                else
                    imeb=ime_b;
                    [xmb,ymb]=center(imeb);
                    brg_e=bwmorph(imeb,'bridge',1);
                    fill_e=imfill(brg_e,'holes');
                end
            end
        end
    end
```

A-4.3 Finding eye brow control points

The selected eye brow image is given as input and its first and last object pixels are returned as initial and final eye brow control point.

```
function [inx,iny,fix1,fiy1]=brow1_con(imag)

figure,imshow(imag);
hold on;
[x,y]=find(imag,1,'last');
plot(y,x,'r.');
inx=y;
iny=x;
[x1,y1]=find(imag,1,'first');
hold on;
plot(y1,x1,'r.');
fix1=y1;
if(x1<iny)
```

```
    fiyl=iny;
else
    fiyl=x1;
end
```

A-5 Eye detection

This module detect eye boundary on the basis of its double derivative and then locate its control points.

A-5.1 Gradient of Eyes

This image takes eye image as input and returned its edge map. Edge map is obtain by taking double derivative f the image and then regional minimum transforms are applied.

```
function [ime,g]=eye_conpt(imag)

im_eyeg=rgb2gray(imag);
%enhancement of image
fil1=fspecial('gaussian');
im_e_g=imfilter(im_eyeg,fil1,'symmetric');
figure,imshow(im_e_g),title('eye con enhance');
hr=fspecial('prewitt');
fs=imfilter(im_e_g,hr,'symmetric');
fs1=fs+im_e_g;
figure,imshow(fs1),title('prewitt');
h=fspecial('sobel');
fd=double(im_e_g);
g=sqrt(imfilter(fd,h,'symmetric').^2+imfilter(fd,h','symmetric').
^2);
s=graythresh(g);
s=s*255; g=g.*255;
BW6 = imextendedmin(g,s);
BW6=~BW6;
ime=BW6;
```

A-5.2 Finding eye control points

Control points of eye are obtained by using its minor and major axis.

```
function [p1,p2,ymean1,l1,l2,xmean1,chk]=eye_con(imag)

 [lo,ro,to,bo]=recsize(imag);
 [l,num]=label_reglips(imag);
 s = struct('im', {});
  for z=1:num
    ch=1;
    [r1,c1]= find(l==z);
    BW2 = bwselect(l,c1,r1,8);
    im2=eye_precon(BW2);
    ar(z)=bwarea(im2);
    if(ar(z)==0)continue; end;
    [lb,rb,tb,bb]=recsize(im2);
    wid(z)=rb-lb;
    hei(z)=bb-tb;
    if((wid(z)==wido) & (heio==hei(z)))
        ch=0; continue;    end;
```

```
    obj = regionprops(double(im2),'all');
    if((ch==1)&(ar(z)>450))
        sol(count)= obj.Solidity
        figure,imshow(im2),title(count);
        s(count).im=im2
        are(count)=ar(z)
        count=count+1;
        if(sol(count-1)<.65) count=1; end;
    end
end
if(count>2)
    maxa=max(are);
    for u=1:count-1
        if(are(u)==maxa)
            im2=s(u).im;
            obj1 = regionprops(double(im2),'all');
            mjaxis1=obj1.MajorAxisLength;
            mnaxis1=obj1.MinorAxisLength;
            orient1=obj1.Orientation;
            [xmean1,ymean1]=center(im2);
            fin_area=obj1.Area
            figure,imshow(im2),title('eye can'); hold on;
            plot(xmean1,ymean1,'g+');
            p1=xmean1-mjaxis1/2;
            p2=p1+mjaxis1;
            plot(p1,ymean1,'r*');
            plot(p2,ymean1,'b*');
            plot(p1:p2,ymean1,'g'); hold on;
            l1=ymean1-mnaxis1/2;
            l2=l1+mnaxis1;
            plot(xmean1,l1:l2,'w');
            plot(xmean1,l1,'w*');
            plot(xmean1,l2,'k*');  hold off;
        else continue; end;
    end
else if(count==2)
        im2=s(1).im;
        obj1 = regionprops(double(im2),'all');
        mjaxis1=obj1.MajorAxisLength;
        mnaxis1=obj1.MinorAxisLength;
        orient1=obj1.Orientation;
        [xmean1,ymean1]=center(im2);
        figure,imshow(im2);hold on;
        plot(xmean1,ymean1,'g+');
        p1=xmean1-mjaxis1/2;
        p2=p1+mjaxis1;
        plot(p1,ymean1,'r*');
        plot(p2,ymean1,'b*');
        plot(p1:p2,ymean1,'g');hold on;
        l1=ymean1-mnaxis1/2;
        l2=l1+mnaxis1;
        plot(xmean1,l1:l2,'w');
        plot(xmean1,l1,'w*');
        plot(xmean1,l2,'k*');
        hold off; end;
end
```

A-5.3 Curve fitting of Eyes

Finally curve is fitted on the extracted control points by using polyfit function.

```
coeffrt = polyfit([llel,lcel,rlel],[lrel,tlel,lrel],2);
xrt_new =llel:.005:rlel;
yrt_new = polyval(coeffrt,xrt_new);
plot(xrt_new,yrt_new,'y');
```

A-6 Lip Detection

This module localizes lips by taking vertical and horizontal projections of the lip zone, then enhancements are applied and its control points are detected.

A-6.1 Vertical and horizontal projection

This function take lower zone of the face as input and returns cropped image by taking its vertical projections (horizontal projects can be taken like wise) and return crop image and top, bottom , left and right control points of the returned image.

```
function [crop_l,lc,tc,wc,hc,crop_bw]=projv_lip(imag)

im=pre_pro(imag);
figure,imshow(im),title('pre pro vp');
[xl,yl]=size(imag);
count=1;i=0;
p=zeros(yl);
for r=1:yl
    for k=1:xl
        if im(k,r)==0
            continue;
        else
            p(r)=p(r)+1;
        end
    end
end
maxv=max(max(p));
sorted=sort(p(:));
meanv=ceil(yl/2);
con=1;
r=meanv;
while(con)
    if (p(r)<maxv-55)|(p(r)==0)  st2=r; con=0; end;
    r=r-1;
end
con=1; r=meanv;
while(con)
    if  (p(r)<mxv-55)|(p(r)==0)  st1=r; con=0; end;
    r=r+1;
end
diff=meanv-st2;
difl=st1-meanv+10;

if(diff<25)
    st2=65; diff=meanv-st2;
end
```

```
if(difl<25)
    stl=220; difl=stl-meanv+10;
end
lc=st2-8; tc=1; wc=diff+difl+8; hc=x1-10;
```

A-6.2 Lip Enhancement

This lip image is then enhanced by using its local statistics that is its mean and variance and enhancement constant is 4.

```
function g=mylocstat2(Icol,M,D,E,k)

%Localize the center of the each columnwise block.
Bcenter=floor((size(Icol,1)+1)/2);
g=Icol(Bcenter,:);
%Preprocess the input parameters.
if (nargin<5) k=[0.4 0.02 0.4]; end;
if (nargin<4) E=4.0; end;
if (nargin<3)
    display('Mean and Variance must be provided!');
end
%Compute the local mean and variance.
Mcol=mean(Icol);
Dcol=std(Icol);
%Build the local response.
resind=find((Mcol<=k(1)*M) & (Dcol>=k(2)*D) & (Dcol<=k(3)*D));
g(resind)=E*Icol(Bcenter,resind);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [bw2]=ftest1(g)

[m n]=size(g);
g=imadjust(g);
g=im2double(g);
fil=1/9*(ones(3));
lap=[0 -1 0; -1 4 -1; 0 -1 0];
smoothim=imfilter(g,fil,'symmetric');
sharpim=imfilter(smoothim,lap,'symmetric');
sharpim=sharpim(1:m, 1:n);
finim=g+sharpim;
h=fspecial('prewitt');
pre=imfilter(finim,h,'symmetric');
finim=finim+pre;
pwrlaw=(finim.*1.05);
bw2=im2bw(pwrlaw,graythresh(finim));
figure,imshow(bw2); title(' Binarized');
bw2=~bw2;
```

A-6.3 Finding lip control points

The control points of the lips are obtain by scanning image first and last object pixel determining left and right control points. And top and bottom control points re detected by moving up and down from center of lips, all these points re returned by this function.

```
function [lplx,lply,lprx,lpry,lptx,lpbx,lpy]=lip_con1(imag)
```

```
[xz yz]=size(imag);
figure,imshow(imag);hold on;
[x,y]=find(imag,1,'first');
plot(y,x,'r.');
[xl,yl]=find(imag,1,'last');
plot(yl,xl,'r.');
ymid=ceil((y+yl)/2);
plot(ymid,x,'r.');
while( imag(z,ymid)~=1) z=z+1; end;
xt=z;
plot(ymid,xt,'r.');
for z=xt+1:xz
    if imag(z,ymid)==1 xb=z; break; end;
    if z==xz xb=xz; end;
end
if (xb<x+5) r=xb-xt; xb=xb+r; end;
plot(ymid,xb,'r.');hold off;
```

A-7 Analysis and Expression recognition:

This module calculate the distances between the control points and compare expression image control points to the normal image and recognize expression on the basis of result of comparison.

```
asple=(lrcpx-llcpx)/(lbcpx-ltcpx);
disletb=(lbcpx-ltcpx);
aspre=(rrcpx-rlcpx)/(rbcpx-rtcpx);
disretb=(rbcpx-rtcpx);
aspl=(flprx-flplx)/(flpbx-5-flptx);
dislrl=(flprx-flplx);
disltb=(flpbx-5-flptx);
meanbi=(inbx1+inbx2)/2
meanbm=(xmb1+xmb2)/2
s=struct('a',browcp1,'b',browcp2,'c',eyelcp,'d',eyercp,'e',lipcp,
'f',asple,'g',disletb,'h',aspre,'i',disretb,...'j',aspl,'k',dislr
l,'l',disltb);
save exp1.mat -struct s;
sn=load('exp1nkh.mat');
browcp1n=sn.b;
eucdistini1=sqrt((browcp1n(2)-browcp1(2))^2+(browcp1n(1)-
browcp1(1))^2)
eucdistmid1=sqrt((browcp1n(4)-browcp1(4))^2+(browcp1n(3)-
browcp1(3))^2)
browcp2n=sn.d;
eucdistini2=sqrt((browcp2n(2)-browcp2(2))^2+(browcp2n(1)-
browcp2(1))^2)
eucdistmid2=sqrt((browcp2n(4)-browcp2(4))^2+(browcp2n(3)-
browcp2(3))^2)
meanbi=(eucdistini1+eucdistini2)/2
meanbm=(eucdistmid1+eucdistmid2)/2
meanasre=(asple+aspre)/2
meanbia=(inbx1+inbx2)/2
meanbma=(xmb1+xmb2)/2
eyelcpn=sn.f;
edlenl=sqrt((eyelcpn(2)-eyelcp(2))^2+(eyelcpn(1)-eyelcp(1))^2)
edlenr=sqrt((eyelcpn(4)-eyelcp(4))^2+(eyelcpn(3)-eyelcp(3))^2)
```

```
edlent=sqrt((eyelcpn(6)-eyelcp(6))^2+(eyelcpn(5)-eyelcp(5))^2)
edlenb=sqrt((eyelcpn(8)-eyelcp(8))^2+(eyelcpn(7)-eyelcp(7))^2)
eyercpn=sn.l;
edrenl=sqrt((eyercpn(2)-eyercp(2))^2+(eyercpn(1)-eyercp(1))^2)
edrenr=sqrt((eyercpn(4)-eyercp(4))^2+(eyercpn(3)-eyercp(3))^2)
edrent=sqrt((eyercpn(6)-eyercp(6))^2+(eyercpn(5)-eyercp(5))^2)
edrenb=sqrt((eyercpn(8)-eyercp(8))^2+(eyercpn(7)-eyercp(7))^2)
meanelcp=(edlenl+edrenl)/2
meanercp=(edlenr+edrenr)/2
meanetcp=(edlent+edrent)/2
meanebcp=(edlenb+edrenb)/2
lipcpn=sn.r;
edlipnl=sqrt((lipcpn(2)-lipcp(2))^2+(lipcpn(1)-lipcp(1))^2)
edlipnr=sqrt((lipcpn(4)-lipcp(4))^2+(lipcpn(3)-lipcp(3))^2)
edlipnt=sqrt((lipcpn(6)-lipcp(6))^2+(lipcpn(5)-lipcp(5))^2)
edlipnb =sqrt((lipcpn(8)-lipcp(8))^2+(lipcpn(7)-lipcp(7))^2)
mean_asp_n=sn.z;
diff_aspre=meanasre-mean_asp_n
jh=s.k-sn.v
hj=s.l-sn.x
if  (((s.k-sn.v)==0)&((s.l-sn.x)==0)&(diff_aspre==0))
    msgbox('Nuetral');
elseif  (((s.k-sn.v)>=23)&((s.l-sn.x)>=5 ...
    &(s.l-sn.x)<=19)&(diff_aspre>=-.27)&(diff_aspre<=.47))
     msgbox('happy');
elseif  (((s.k-sn.v)>=-10)&((s.k-sn.v)<=8)&((s.l-sn.x)>=-9 ...
    &(s.l-sn.x)<12)&(diff_aspre>=.325)&(diff_aspre<=.71))
    msgbox('sad');
elseif  (((s.k-sn.v)>=-11)&((s.k-sn.v)<=12)&((s.l-sn.x)>=0 ...
    &(s.l-sn.x)<19)&(diff_aspre>=-.12)&(diff_aspre<=.58))
    msgbox('fear');
elseif  (((s.k-sn.v)>=-6)&((s.k-sn.v)<=15)&((s.l-sn.x)>=21)...
        &((s.l-sn.x)<=45)&(diff_aspre>=-
.48)&(diff_aspre<=1.02)...
        &(aspl>=-1.9)&(aspl<=1.75))
    msgbox('surprise');
elseif  (((s.k-sn.v)>=-15)&((s.k-sn.v)<=6)&((s.l-sn.x)>=-15& ...
        (s.l-sn.x)<=12)&(diff_aspre>=-.04)&(diff_aspre<=.33))
        msgbox('Anger');
elseif  (((s.k-sn.v)>=-7)&((s.k-sn.v)<=26)&((s.l-sn.x)>=-14& ...
        (s.l-sn.x)<=25)&(diff_aspre>=.56)&(diff_aspre<=.71) ...
        &(aspl>=-1.18)&(aspl<=3.6))
        msgbox('disgust');
else
        msgbox('other');
end
```

References

References

[1] Arif Zuberi, Muhammad Ijaz Khan and Ayesha Farrukh, **"Automatic Spatial Segmentation of Facial Regions in HSI Color Space"** *2003.*

[2] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain, **"Face Detection In Color Images"** *2002.* *http://www.cg.cs.unibonn.de/docs/teaching/2002/WS/cv_hand_tracking/documents/papers/face-detection-in-color.pdf.*

[3] Sanjay Kr. Singh1, D. S. Chauhan2, Mayank Vatsa3, Richa Singh3*1, **"A Robust Skin Color Based Face Detection Algorithm"**, *Tamkang Journal of Science and Engineering, Vol. 6, No. 4, pp. 227-234 (2003)* http://www2.tku.edu.tw/~tkjse/6-4/6-4-6.pdf

[4] W. K. Teo1, Liyanage C De Silva2 and Prahlad Vadakkepat1, **"Facial Expression Detection And Recognition System"**, *Journal of The Institution of Engineers, Singapore Vol. 44 Issue 3 2004* http://www.ies.org.sg/journal/past/v44i3/v44i3_2.pdf.

[5] Ying-li Tian and Ruud M. Bolle, **"Automatic Detecting Neutral Face for Face Authentication and Facial Expression Analysis"**, *issue24 march 2003* http://www.research.ibm.com/peoplevision/Neutralface.pdf

[6] Maja Pantic, Member, IEEE, and Leon J. M. Rothkrantz, **"Facial Action Recognition for Facial Expression Analysis From Static Face Images"**, *IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 34, No. 3, June 2004* http://mmi.tudelft.nl/pub/maja/_SMCB-2004-final.pdf.

[7] D. Datcu, L.J.M. Rothkrantz, **"Automatic recognition of facial expressions using Bayesian Belief Networks"**, *issue 2004* http://mmi.tudelft.nl/pub/dragos/Automatic%20recognition%20of%20facial%20expressions%20using%20BBN.pdf

[8] Ying-li Tian, **"Evaluation of Face Resolution for Expression Analysis"**, *issue 2004* http://www.visioninterface.net/fpiv04/pdf/21fa.pdf.

[9] Leonardo Trujillo, Gustavo Olague, Riad Hammoud and Benjamin Hernandez, **"Automatic Feature Localization in Thermal Images for Facial Expression Recognition"**, *Issue 2005*

[10] Ashutosh Saxena, Ankit Anand, Prof. Amitabha Mukerjee,**"Robust Facial Expression Recognition Using Spatially Localized Geometric Model"** *International Conference on Systemics, Cybernetics and Informatics, Issue 2004.* http://www.stanford.edu/~asaxena/resources/papers/icsci2004_facial.pdf.

[11] Ioulia Guizatdinova and Veikko Surakka **"Detection of Facial Landmarks from Neutral, happy, and Disgust Facial Images",** *Research Group for Emotions, Sociality, and Computing Tampere Unit for Computer-Human Interaction Department of Computer Sciences University of Tampere,Finland. Issue 2004* http://wscg.zcu.cz/wscg2005/Papers_2005/Full/K59-full.pdf

[12] **"Digital Image Processing";** Rafeal Gonzalez, Richard Woods; Third Edition; Published by Pearson Education, Singapore.

[13] **"Practical Computer Vision Using C"** J.R. Parker, ISBN 0-471-59262-5.

[14]" **Pattren Recognition: Statistical, Structural and Neural Approaches";** Robert J. Schalkoff; First Edition; Published by John Wiley & Sons Inc.1992.

[15] **"Digital Image Processing with MATLAB";** Rafeal Gonzalez, Richard Woods; Third Edition; Published by Pearson Education, Singapore.

[16] http://mathworld.wolfram.com/EquilateralTriangle.html

[17] http://mathworld.wolfram.com/Distance.html

[18] http://www.answers.com/main/ntquery?dsid=2222&dekey=HSV+color+space&Linktext =HSV%20color%20space

Dr. Syed Afaq Hussain

From: <info@icitst.org>
To: <drafaq@iiu.edu.pk>
Cc: <c.shoniregun@infonomics.org.uk>
Sent: Sunday, August 20, 2006 7:38 PM
Subject: ICITST-2006 Status of paper 47

Dear Afaq Husain, Sadia Arshid, Maham Javed,

This is a reminder with regards to your conference registration. Our current list shows that your registration fees are still pending. Please note that if your registration is not received by 28 August 2006 we will not be able to include your paper in the proceedings.

Paper: 'Automatic Facial Expression Recognition'
Author(s): Afaq Husain, Sadia Arshid, Maham Javed

Can you please make sure that you post the original signed copy of the registration form (make it a recorded delivery) to:

Dr C.A.Shoniregun (ICITST-2006 general chair)
School of Computing & Technology
University of East London
Longbridge Road
Dagenham, Essex
RM8 2AS
United Kingdom.

We will send you the reviewers' comments and the final paper submissions guidelines immediately after we have received the original registration form. Once received, you will be expected to incorporate the changes within two weeks.

As soon as we receive the original registration from, we will be able to issue an official invitation letter.

Please kindly contact ICITST-2006 to let us know the state of your participation in the conference.

Best regards,

Dr Charles A. Shoniregun
ICITST-2006 General Chair

Automatic Facial Expression Recognition

Syed A. H., Sadia A., Maham J.
drafaq@iiu.edu.pk, cutecomrade@yahoo.com

Abstract
One of the major areas of current research is intelligent computers. Research is focusing on making Intelligent Computers that can understand human voice, natural language, and even faces and patterns. Emotions play a vital role in depicting human behaviour. Emotions are mostly revealed through expressions. In this paper we present our research on the detection and recognition of facial expressions by a computer. This paper presents the technique of facial gesture recognition for the static images based on feature extraction. Six universal expressions are considered: happy, sad, fear, surprise, anger and disgust with reference to the neutral face. Face and its deformable features (Eyes, mouth) are localized on the basis of color component. We have adopted multiple techniques for detection of various features, e.g., eye brows are extracted by taking edges, eyes are extracted by using gradient and lips are extracted by taking integral projections of the green plane. Various control points have been identified and changes in these control points help in recognition of the expressions. The experimental results show 94% accuracy in expression classification of our system.

Keywords: Expression Recognition, Pattern Recognition, Human Computer interaction, Affective computing

1. INTRODUCTION

Intelligent human computer interaction is the area of considerable research for the past few years. One way to achieve this goal is making computers intelligent enough so that they interact with users in the same way as humans interact with each other.

Humans naturally interact with each other through verbal and non verbal sign systems. Major part of communication is conveyed non-verbally. Considering nonverbal communication it is possible to say that face expressions exhibit about half of the transmitted notions [Iou04]. Facial expressions play a significant role in our social and emotional lives.

They are visually observable, conversational, and interactive signals that clarify our current focus of attention and regulate our interactions with the environment and other persons in our vicinity [Maj04].

The analysis and recognition of facial expression comes under the field of **"Affective computing"** which is recognizing understanding and artificially exhibiting emotions to some extent.

Facial appearance of a person without any dramatic turn of phrase is a neutral face and most of the time a person has a neutral face. Deformations of the neutral face due to psychological effects form expressions. Deformations can be either contraction or expansions of the facial muscles.

Humans have ability to recognize expressions without any effort or delay but replicating this ability in computers is still a challenging problem because expressions are generated by non-rigid object deformations which vary from person to person. Moreover facial features vary from person to person and these are also highly deformable. Computerized process of understanding expression comprises location of features, analyzing them and then classifying them.

Face detection is a preliminary step for facial feature detection. Face is detected on the basis of skin tone color then it is divided into two zones; the eye zone and the lip zone. Features to be extracted for expression classification are eyebrows, eyes and lips. The control points of these features are then given to the statistical analyzer which classifies expression in comparison with neutral face. Neutral face is taken as a reference to overcome divergence in facial expressions.

2. PREVIOUS WORK

A lot of research work has been carried out in the area of expression recognition. Various techniques for facial expression recognition can be broadly divided into two classes; feature based approaches and probabilistic approaches. Probabilistic approaches do not give preference to the facial features such as eyes, mouth. Instead, the feature vector can be random distribution of image intensities and these vectors may differ for each emotion. These vectors are calculated per emotion and classification algorithm like Hidden Markov Models, Neural net or Hybrid approaches can be applied [Teo04]. For feature based approaches, the characteristics of features and their shapes are important in expression classification.

W. Teo et al, [Teo04] have used integral projections for feature extraction. Optical flow information of these features is given to neural nets for expression classification into four expressions. Carmen Frank et al [Car03] have optimized Eigen faces by face masks for classification of anger and neutral face. D. Datcu et al, [Dat04]] have devised the system of automatic facial expression classification by using Bayesian belief neural nets. Features are extracted in form of Action Units using FACS. Jun Ohya et al, [Jun98] has suggested that change/ motion in facial feature can be observed through DCT constants. R.Q. Fetiosa [Fet00] also uses back propagation neural networks having radially symmetric activation functions to classify facial expressions and feature extraction is done through PCA. Petar S. Alksic [Pet05] used multi-streams Hidden Markov Model system and facial action parameters (FAP) have been used for feature extraction. The technique proposed by Ying li Tian [Yan03] comprises fisher discriminants for facial feature extraction. Gabor wavelets have also been used for feature extraction [Hai04]. Ashutosh et al, [Ash04] created a spatio-temporal representation of the face and based on geometrical relationship between features Euclidean distances were measured and fed into feed forward neural network. Facial landmarks can also be detected for facial expression analysis [Iou04]. Maja Pantic [Maj04] has worked with static images in dual view planes using facial action coding system classifying the expression using rule based classification method.

Most of the work on facial expression recognition has been done on the video streams, which are easy to work with, as temporal dynamics information is available while determining the expression in static images is quite difficult. The general techniques commonly classify three to four expressions only. Feature extraction is mostly done using the intensity images although it may be more difficult to extract facial features in gray level images, because the characteristics of skin tone color are not available. The literature survey also reveals that complex techniques such as neural networks, SVMs*[1], Hidden Markov models or rule based classifies are used for classification purpose.

[1] SVM: Support Vector Machines

3. PROPOSED METHODOLOGY

In this paper we present a method for automatic facial expression recognition for static images. The system recognizes six universal expressions that are happy, sad, disgust, fear, surprise and anger in comparison with the neutral face. The templates for these expressions are given in figure 1.

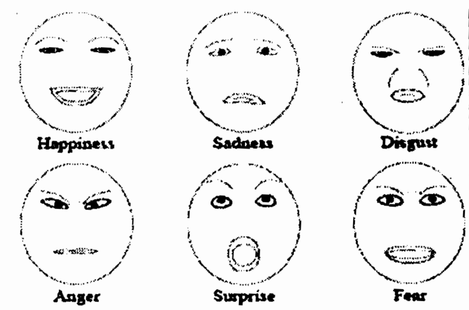


Figure1: The six universal expressions [Yan03]

The feature extraction has been done using the color characteristics. Statistical analysis of the Euclidean distances between expressional and expressionless face has been utilized for feature analysis. The block diagram for the proposed model is given in figure 2.

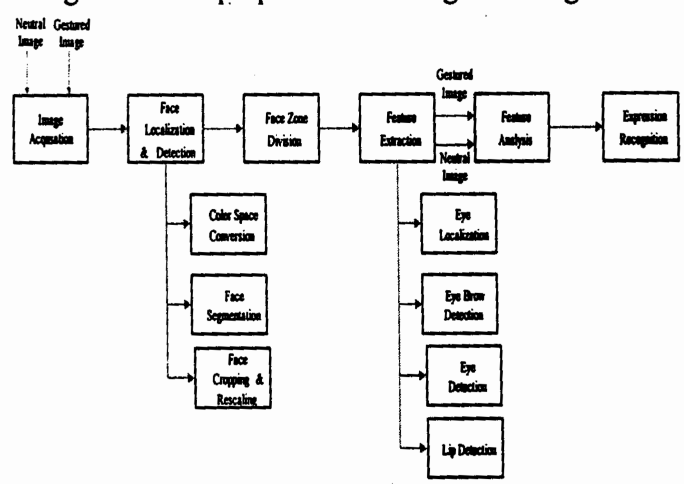


Figure 2: Block Diagram of Proposed System

The constraints for the system are that there should be one face with frontal view in the well illuminated image. The person should not have beard and mustaches and he should be without glasses. The modules of the system are as follows:

3.1 Face localization

The localization of the face is foremost step in expression recognition. The goal of face localization is to spot the face in the image and returning its exact location. Many of researches have been done on face localization in color images using different color spaces as YCbCr, RGB, HSV [San03]. Face segmentation in this research has been done in HSV color space due to its initiative nature. Specific range of hue determines skin values and pixels lying in this range retain their values which help in background elimination. On the basis of area and eccentricity, candidate face is selected. Orientation of the face is corrected if required and ellipse is fitted around localized face. Face region is cropped and resampled to the resolution of 350 X 290 for further processing.

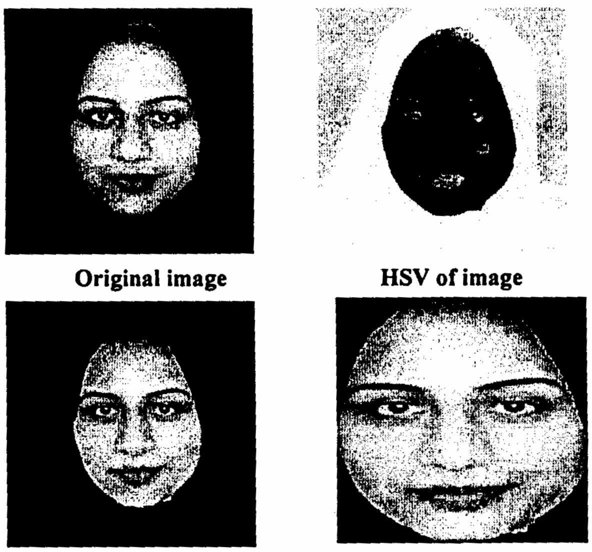


Figure 3: Face localization

3.2 Face zone division

After resampling, face is divided into two zones: upper zone containing eye, and lower zone containing rest of the face on the basis of its major and minor axis as given in the figure4. Zone division reduces the search space for feature extraction.

Figure 4: Face divided into zones

3.3 Feature extraction

The elements participating in the face expressions are lips, eyes, eye brows, cheek bones, forehead wrinkles, nose etc. but the features required for this research are eye brows, eyes and lips. For each feature different detector is used, as the properties (shape, color) of features vary from one another. For eye brows and eye extraction, eye localization s required.

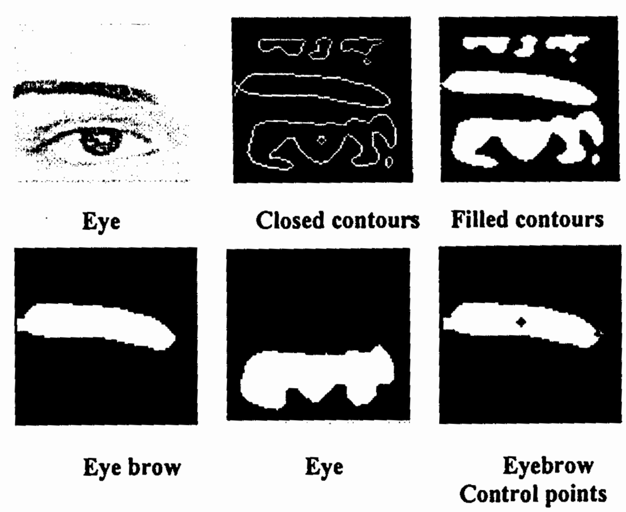
a. Eye localization

Eyes are localized on the face that eyes does not belong to skin. Non skin regions are determined on the basis of hue in the upper face zone. The resultant blobs are then checked for consistent area and horizontal alignment. Moreover distance between the eyes should be in specific threshold to determine blobs as eye candidates. Bounding boxes are then drawn on the candidate eyes.

Figure 5: Localized eyes

b. Eye brow Detection

Edge detection technique is used for eye brow detection. The technique is based on the fact that edges of the eye brows and eyes form closed contours which are found by using LOG*[2] with zero thresholds as proposed by Aushutosh [Ash04]. This technique is enhanced by using standard deviation ranges from 4 ~5.2 to have more accurate contours. The contours are then filled [Ash04] .Morphological filters are applied on the resulting image using disk shaped structuring element. Undesired objects are rejected on the basis of area and out of two largest objects; one with less centroid is selected as eye brow candidate.



---

[2] LOG Laplacian of Gaussian filter

c. Eye Extraction

After eye brow detection eye region is further cropped resulting only in eye. Gradient is applied on the image and eye edge map is obtained by applying regional minimum transforms. Morphological operations are applied to obtain proper eye boundary and major and minor axis are used for its control point detection, and curve fitting is applied finally.

Figure 6: Eye brow detection

d. Nose Estimation

Once the eyes are being detected, the nose tip is estimated by visualizing the fact that nose tip and center of both eyes form equilateral triangle. So the eye middle points are estimated. In this case we have base of the triangle. Its height is calculated by using formula

$$h = \alpha \sin 60^\circ = \frac{1}{2}\sqrt{3}\alpha$$

Where 'α' is the length of side.

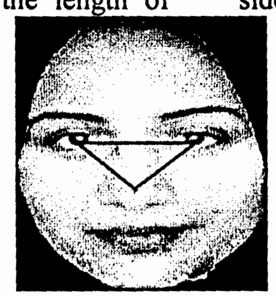


Figure 7: Estimated nose

e. Lip localization

For lip localization, lower zone of the face is decomposed into its red, green and blue component. Green component is then used for further processing, because of its sensitivity towards luminance. Horizontal and vertical projections are used to reduce search space [Teo04].This image is then enhanced by applying image adjustment based on local statistics for sliding block operation. Then morphological operations are applied and the maximum region is selected as the lip candidate. The top, bottom, left and control points are then detected.

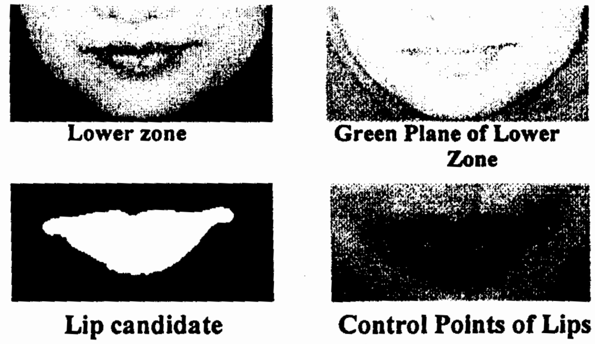


Figure 8: lip localization

3.4 Feature Analysis

The control points required for analysis are initial and middle control points of eyebrow, four control points of eye that is middle point of upper and lower eyelid and two extreme corners of eye, four lip control points which are left, right, top and bottom control points of lips. These distances are shown in figure 9 as Distances d2 width of mouth, d1 distance between extreme points of the lips, d3 distance between upper and lower eyelid are calculated.

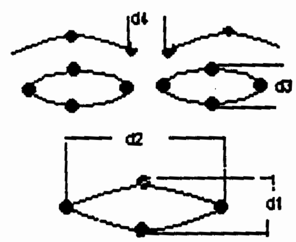


Figure 9: Distance Transform

The Euclidean distances of these control points are taken with reference to neutral face. The Euclidean distance is given by

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Aspect ratio of eyes and lips in gestured image are calculated and its difference with the aspect ratio of eyes and lips of neutral image is taken respectively. Statistical analysis of these distances classifies the gestured image into the corresponding expression.

4. RESULTS

The experimental results have shown that expression classification rate is about 94% which is quite satisfactory. The testing set contained 35 images with 5 samples for each expression. Classification results of AFER system given in table 1 show that the performance of our AFER system is 100% in case of happy, disgust and fear. For surprise and sad it is 80% and performance for anger is 75%.

Expression/Rates	Correct Classification Rate	Wrong classification rate
Happy	100%	0%
Sad	80%	20%
Anger	75%	25%
Disgust	100%	0%
Surprise	80%	20%
Fear	100%	0%

Table 1: Classification Rate of Expressions

Confusion matrix in table 2 represents that one of the anger expression is determined as fear, because overlapping boundary of these two expressions make learning a difficult problem. Moreover, for statistical analysis a large data set is required for better results so the performance of the system in some cases was degraded because of limited data set. Wrong classifications are also due to the reason that expressions were not clearly portrayed by the subjects who could not act well to express the assigned expressions.

Expressions	Happy	Sad	Anger	Disgust	Surprise	Fear
Happy	4	0	0	0	0	0
Sad	0	4	0	0	0	0
Anger	0	0	3	0	0	1
Disgust	0	0	0	4	0	0
Surprise	0	0	0	0	4	0
Fear	0	0	0	0	0	4

Table 2: Confusion Matrix

(a) Sad (b) Happy (c) Surprise
Figure 10 Expressions correctly classified

(a) Fear (b) Anger (c) Disgust
Figure 11 Expressions correctly classified

(a) Sad Unrecognized (b) Disgust unrecognized
Figure 12 some wrong classifications

5. Concluding Remarks

Automated analysis of faces showing different expressions has been studied to improve the quality of human-computer interaction. The ability of a computer to detect, analyze and, finally, recognize a user's face has many applications in the domain of HCI. The goal of our work was to recognize six basic expressions that are happy, disgust, sad, surprise, fear and anger in static images by using the configurational details of them.

The technique we have used is efficient in terms of time and space. AFER is scale, rotation and translation invariant and we have done color based feature localization.

6. Future enhancements

Future areas of research can include expression classification in multi face images. Spectacled faces can be considered. Expressions of the subject having beard and mustaches can also be classified. The technique can be enhanced by using complex background instead of using uniform background. The enhancement can be made by applying a different classification technique like neural networks, SVMs, etc. As the technique is efficient in terms of time and space so real time application can be made.

REFERENCES

[Teo04] W. K. Teo1, Liyanage C De Silva2 and Prahlad Vadakkepat1, "Facial Expression Detection and Recognition System", *Journal of the Institution of Engineers, Singapore Vol. 44 Issue 3 2004*

[Car03] Carmen Frank and Elmar N¨oth "Optimizing Eigenfaces by Face Masks for Facial Expression Recognition" *Lehrstuhl f¨ur Mustererkennung, Universit¨at Erlangen-N¨urnberg, August 2003, CAIP Groningen, NL*

[Dat04]] D. Datcu, L.J.M. Rothkrantz, "Automatic recognition of facial expressions using Bayesian Belief Networks", *issue 2004*

[Jun98] Jun Ohya, "Face/Gesture Analysis/Synthesis Technologies for Human too-Human Communications through Virtual Environments" *issue 1998*

[Fet00]R.Q.Feitosa M.M.B.Vellasco, D.T.Oliveira, D.V.Andrade, S.A.R.S.Maffra, "Facial Expression Classification Using RBF AND Back-Propagation Neural Networks" *issue 2000*

[Pet05] Petar S. Aleksic and Aggelos K. Katsaggelos, "Automatic facial expression recognition using facial animation Parameters and multi-stream HMMs", *issue 2005*

[Yan03] Ying-li Tian and Ruud M. Bolle, "Automatic Detecting Neutral Face for Face Authentication and Facial Expression Analysis", *issue24 march 2003*

[Hai04]Haisong Gu, Qiang Ji, "Information extraction from image sequences of real-world facial expressions" *issue 2004*

[Ash04] Ashutosh Saxena, Ankit Anand, Prof. Amitabha Mukerjee,"Robust Facial Expression Recognition Using Spatially Localized Geometric Model" *International Conference on Systemics, Cybernetics and Informatics, Issue 2004.*

[Iou04] Ioulia Guizatdinova and Veikko Surakka "Detection of Facial Landmarks from Neutral, happy, and Disgust Facial Images", *Research Group for Emotions, Sociality, and Computing Tampere, Unit for Computer-Human Interaction, Department of Computer Sciences University of Tampere, Finland. Issue 2004.*

[Maj04] Maja Pantic, Member, IEEE, and Leon J. M. Rothkrantz, "Facial Action Recognition for Facial Expression Analysis from Static Face Images", *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 34, No. 3, June 2004.*

[San03]Sanjay Kr. Singh1, D. S. Chauhan2, Mayank Vatsa3, Richa Singh3*1, "A Robust Skin Color Based Face Detection Algorithm", *Tamkang Journal of Science and Engineering, Vol. 6, No. 4, pp. 227-234 (2003)*