Face Detection using Bacteria Foraging

Major Project submitted in partial fulfillment of the

Requirements for the award of degree of

Master of Technology

Information Systems

Submitted By:

Jitendra Kumar Niranjan

(04/IS/09)

Under the Guidance of:

Prof. O. P. Verma

(HOD, I.T. Department)

Department of Information Technology

Delhi Technological University

New Delhi, 2011

CERTIFICATE

This is to certify that **Mr. Jitendra Kumar Niranjan (04/IS/09)** has carried out the major project titled "**Face Detection using Bacteria Foraging**" as a partial requirement for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2009-2011**.

The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

Prof. O.P. Verma
Head of Department
Department of Information Technology

Delhi Technological University
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I express my gratitude to my major project guide ***Prof. O.P. Verma***, HOD, IT Dept., Delhi Technological University, for the valuable support and guidance he provided in making this project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Jitendra Kumar Niranjan
04/IS/09
M.Tech. (Information System)
Department of IT

E-mail: jituniranjan2000@gmail.com

Abstract

Face detection is a challenging computer vision problem. Given a still image or an image sequence, the goal of face detection is to locate all regions that contain a face regardless of any three dimensional transformation and lighting condition. Several independent research have proposed different methods addressing the problem of face detection such as Principal Component Analysis, Linear Discriminant Analysis, Elastic Brunch Graph Matching, etc. Many approaches have been proposed using the skin color Information and shape analysis. There are two main categories that may serve as a solution for this problem: feature-based and image-based approaches.

In this report, we present a detailed experimental study of face detection algorithm based on human skin color and fitting the ellipse as human face can be approximate by ellipse. Our approach for face detection requires no initial pre processing of the image. We proposed a face detection method based on Bacteria foraging, an optimization technique. A number of Bacteria agents with evolutionary behaviors are uniformly distributed in the 2-D image environment to search the skin-like pixels and locate each face-like region by evaluating the local color distribution.

Table of Content

Chapter 1

INTRODUCTION

1.1 Biometric Systems

Identity management refers to the challenge management of providing authorized users with secure and easy access to information and services across a variety of networked systems. A reliable identity management system is a critical component in several applications that provide services to only legitimately enrolled users. Examples of such applications include physical access control to a secure facility, access to computer networks, performing remote financial transactions etc. The primary task in an identity management system is the determination of individual's identity. This action may be necessary for many reasons but in most applications, the primary intention is to prevent impostors from accessing protected resources. Traditional methods of establishing a person's identity include knowledge-based like passwords and token-based like ID cards, but these representations of the identity can easily be lost, shared or stolen. Therefore, they are not sufficient for identity verification. Biometric offers a natural and reliable solution to the problem of identity determination by recognizing individuals by using certain physiological or behavioral traits associated with the person.

Some of physiological and behavioral characteristics that are being used in biometric systems include finger prints, iris, face, ear, signature, voiceprint, palmprint etc. They have an advantage over traditional security methods in that they cannot be easily stolen or shared. A simple biometric system has five main modules:

1. Sensor module
2. Feature extraction module
3. Matching module
4. Decision module
5. System database module

The sensor module is responsible for acquiring the biometric data from an individual. The feature extraction module processes the acquired data and extracts a set of salient discriminatory feature to represent the underlying trait. The matching module compares extracted features against the stored templates to generate match scores. The decision module uses the match scores to either validate a claimed identity or determines the user's identity. The system database module acts as the repository of biometric information.

1.2 Operations of a Biometric System

Depending on the application context; a biometric system may operate in the following two modes:

1. Verification
2. Identification

1.2.1 Verification

In the verification mode, the system validates a person's identity by comparing the captured biometric data with his own biometric template stored in the system database. In such a system, an individual who desires to be recognized claims an identity, usually via a PIN, a user name or a smart card and the system conducts a one to one comparison to determine whether the claim is true or not. The verification problem is two-category classification problem as follows:

Given an input query feature set X and a claimed identity I, determine if(X,I) belongs to T or F, where T indicates that the claim is true (genuine user) and F indicates that the claim is false (impostor). To determine its category, X is matched against Y, the stored biometric template of user I. The resulting decision rule is,

$$(X,I) = \begin{cases} T & If\ S(X,Y) \geq Th \\ F & Otherwise \end{cases} \quad \textbf{(1)}$$

Where S is the function that measures the similarity between X & Y, and Th is a predefined threshold. The value S(X, Y) is a match score between the feature vector of the query and the stored template corresponding to identity I.

1.2.2 Identification

In the identification mode, the system recognizes an individual by searching the templates of all the users in the database for a match. Therefore, the system conducts a one to many comparisons to establish an individual's identity or fails if the subject is not enrolled in the system database. The subject does not claim any identity. The identification problem may be stated as follows:

Given an input query feature set X , determine the identity I_k , k Є {1,2,.....,N, N+1}, where I_1, I_2, ..., I_N are the N identities enrolled in the system, and I_{N+1} indicates the reject case. To determine the individual's identity, the decision rule is:

$$X \in= \begin{cases} I_M & If\ M = max_k\{S(X, Y_{I_k})\}\ and\ S(X, Y_{I_M}) \geq Th \\ I_{M+1} & Otherwise \end{cases} \quad \textbf{(2)}$$

Where S is the function that measures the similarity between X and $Y_{Ik,}$ $Y_{Ik}$is the biometric template corresponding to identity $I_{k,}$ and Th is a predefined threshold.

The value $S(X,Y_{ik})$ is a match score between the feature vector of the query and the stored templates corresponding to identity I_k.

The above described identification is the open-set identification. Another one is closed set identification in which the user is known to exist in the database. There is never a reject case in the closed set identification.

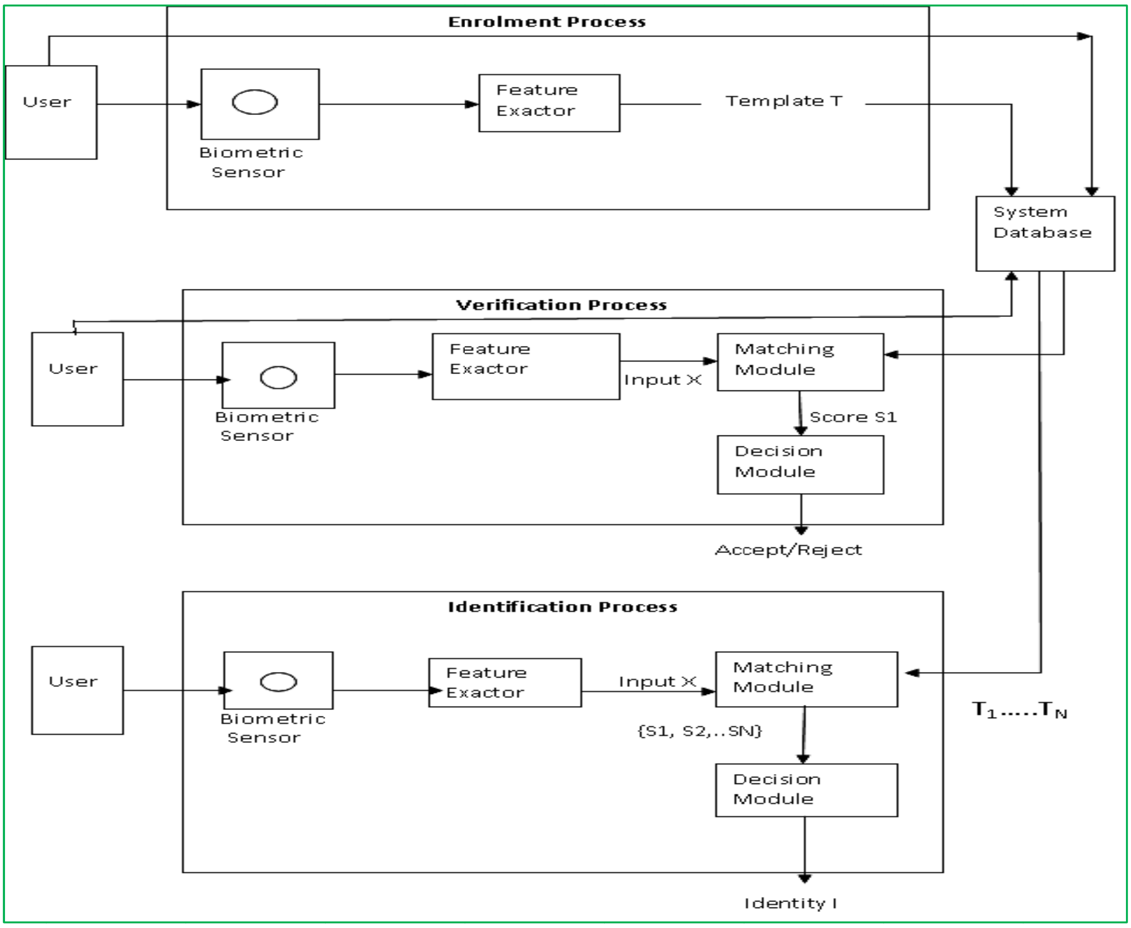


Fig.1. Enrolment, Verification and Identification Process.

1.3 Performance of a Biometric System

A match score is known as genuine score if it is a result of matching two samples of a biometric trait of the same user. It is known as an impostor score if it is the result of matching two samples of a biometric trait originating from different users. An impostor score that exceeds the predefined threshold results in a false accept, while a genuine score that falls below the predefined threshold results in a false reject. The False Accept Rate (FAR) of a biometric system is the fraction of impostor scores exceeding the threshold. Similarly, the False Reject Rate (FRR) of a system is defined as the fraction of genuine scores falling below the threshold. Regulating the value of Threshold changes the FRR and the FAR values, but for a given biometric system, it is not possible to decrease both these errors simultaneously. The EER (Equal Error Rate) refers to the point where the FAR equals the FRR. A lower EER value indicates better performance. In case of identification, an alternative performance measure called identification rate is used to indicate the performance of the system. The identification rate indicates the proportion of times a previously enrolled user is successfully mapped to the correct identity in the system. The rank-k identification rate is the proportion of times the correct identity occurs in the top k matches as determined by the match score.

1.4 Limitations of Biometric System

Biometric Systems are often affected by the following problems:

- Noise in sensed data: The recognition accuracy of the biometric system is highly sensitive to the quality of the biometric input and noisy data can result in a significant reduction in the accuracy of the biometric system.

- Non-universality: If every individual is able to present the biometric trait for recognition, then the trait is said to be universal. Non-universality leads to Failure to Enroll (FTE) error in a biometric system.
- Lack of individuality: Feature extracted from different individuals can be quite similar. This lack of uniqueness increases the False Accept Rate (FAR) of a biometric system.
- Intra-class variations: The biometric data acquired during verification will not be identical to the data used for generating template during enrollment for an individual. This is known as intra-class variation. Large intra-class variations increase the False Reject Rate (FRR) of a biometric system.
- Inter-class similarities: Inter-class similarity refers to the overlap of feature spaces corresponding to multiple individuals. Large Inter-class similarities increase the False Acceptance Rate (FAR) of a biometric system.
- Susceptibility to circumvention: An impostor can circumvent a biometric system using spoofed traits. Behavioral traits like signature and voice are more susceptible to such attacks than physiological traits.

Chapter 2

FACE DETECTION

2.1 Introduction

Human face perception is currently an active research area in the computer vision community. Face detection falls into the field of *computer vision and pattern recognition*. This field seeks to use the computer as a means of observation, attempting to make it a tool for perceiving and processing visual information in addition to the more orthodox textual use of computers. Within the studies of Artificial Intelligence this field is of great importance, since perception is an important task to be accomplished by Intelligent Autonomous Systems, such as robots, to be able to react (and perhaps adapt) to their surroundings. Face Detection is a very important task within the field of computer vision, for it is the first essential step in a complete *face recognition* system. Furthermore it has its applications in security systems, multimedia systems and human-computer interaction systems, i.e. a computer monitor flashing on, only when a person is in front of it, and going into power saving mode when the person leaves. Human face localization and detection is often the first step in applications such as video surveillance, human computer interface, and image database management.

Locating and tracking human faces is a prerequisite for face recognition and facial expressions analysis, although it is often assumed that a normalized face image is available. In order to locate a human face, the system needs to capture an image using a camera and a frame-grabber to process the image, search the image for important features and then use these features to determine the location of the face. The process of human face detection is a complex problem to computer but if sufficient information provided, it will be able to detect human face. One of the most important features for detecting the face is the skin color of human face. However color is not sufficient to detect the human and localize it because other body part may have similar color. Therefore we need some other features such as geometry of the face and eyes location.

Many Face detection techniques are reported in the literature. Face detection in gray images based on edge orientation feature have proposed by [1].Another technique has been proposed by [2] in color image using the shape of face and edges. Johg MinLee et al[3] has worked on face detection suing edge orientation and geometric feature.Wu and Nevatia et al [4]present an approach to automatically detect and track multiple, possibly partially occluded humans in a walking or standing pose from a single camera, which may be stationary or moving. Jinquinglui et al. [5] suggest an improved canny and soble edge operator for the face detection. The Kirsch edge detection algorithm's shortage, a method of improved Kirsch human face edge detection was proposed by Yuweibo et al. [6]. Many researchers have worked on face detection using the template matching. Ben Zion et al [7] investigate the problem of detecting human faces in complex images using large set of templates and hash table for improving the searching efficiency. [8, 9] have proposed new method for the face detection using the template matching. Jianguo et al [10] propose a new method based on shape information and edge template matching. A classical example of appearance-based methods, Mogghadametal. [11] Used Eigen- space representation for individual features, i.e. the eyes and mouth. However, this method was also effected by rotation and illumination. Another category is the invariant feature-based approach. These methods attempt to find consistent features that are discriminative enough even when the pose, viewpoint, or illumination conditions could vary. Among many approaches involving the invariant-based method for facial features, the facial color and the edge property are two typical invariants. The facial color characteristic is a simple and fast approach for pose-variant face detection. In [12, 13], which use facial color information with lighting compensation, a chromatic map of the facial features is created. Taigun Lee et al. [14] proposed new method based on pattern of facial feature in color images.

Kinchoong et al. [15] suggest new methods which adopt a bottom-up feature-based approach which has the flexibility to be extended to different scale, orientation and viewpoint of faces in the image. The neural network approach detects faces by sub-sampling different regions of the image to a standard- sized sub image and then passing it through a neural network filter. Some authors [16, 17, 18, 19] use artificial neural network trained with the normalized face image but these method requires more computation time. Generative neural network model have been proposed by Raphael Feraud et al. [20] based on neural network. Another category of face detection is using fuzzy pattern matching and neuro-fuzzy classifier. Haiyuan et al. [21] proposed a fuzzy theory based on pattern matching technique and use it to detect a face candidate finding out pattern similar to prebuild head shape model from the extracted skin and hair region. Akihiro yorita et al. [22] suggests a fuzzy based using the genetic algorithm an optimization technique. Recently, evolutionary computation has been applied to improve the performance of image processing [23, 24].

2.2 Problem definition

Face detection remains an open problem. The task of face detection is extremely trivial for humans, but it is a challenge to enable computers to carry out the same task. The problem of face detection in still images is more challenging and difficult when compared to the problem of face detection in video since emotion information can lead to probable regions where face could be located. We may summarize the basic problem definition of Face Detection by saying that we are given an input scene. The goal is to find a set of possible locations on the image satisfying certain conditions. We are subject to the constraint that we are able to find the faces from the scene in a short period of time. We have a diverse database of color images, and we want to determine whether these images contain any human faces. A lot of literature is written on the subject, and a lot of approaches to the face detection problem are proposed.

2.3 Objective and Overview of the Project

The main objective of our project was to study existing techniques for face detection, and propose changes or modification for improvement of current methods. Our work involves initial extraction of regions of the image based on skin color range, using bacteria foraging. Minimizing false positives and time required were two constraint imposed. Since our approach is rather open, it can be implemented in a variety of platforms and a multitude of applications. All our simulations have been done in a testing and simulating tool developed in MatLab.

2.4 Related works on Face Detection

Many researchers have proposed different methods addressing the problem of face detection. Face detection techniques are classified in to feature based and image based. The feature based techniques use edge information, skin color, motion and symmetry measures, point distribution, et al [25]. Image based techniques include neural networks, linear subspace method like Eigen faces, fisher faces etc. A number of face detection algorithms such as those using eigenfaces [26] and neural networks [27], for instance, have been developed. In these algorithms, however, a large amount of numerical computation is required, making the processing extremely time-consuming.

2.4.1 Edge-based Feature Maps

Edge-based feature maps are the very bases several image representation algorithm [28], [29]. The feature maps represent the distribution of four- direction edges extracted from a 64x64 pixel input image. The input image is first subjected to pixel by pixel spatial filtering operations using kernels of 5x5 pixel size to detect edges in four directions, i.e. horizontal, +45 degree, vertical, or -45 degree. The threshold for edge detection is determined taking the local variance of luminance data into account. Namely, the median of the 40 values of neighboring pixel intensity differences in a 5x5 pixel kernel is adopted as the threshold. This is quite important to retain all essential features in an input image in the feature maps. Fig.2 shows an example of feature maps generated from the same person under different illumination conditions. The edge information is very well extracted from both bright and dark images.

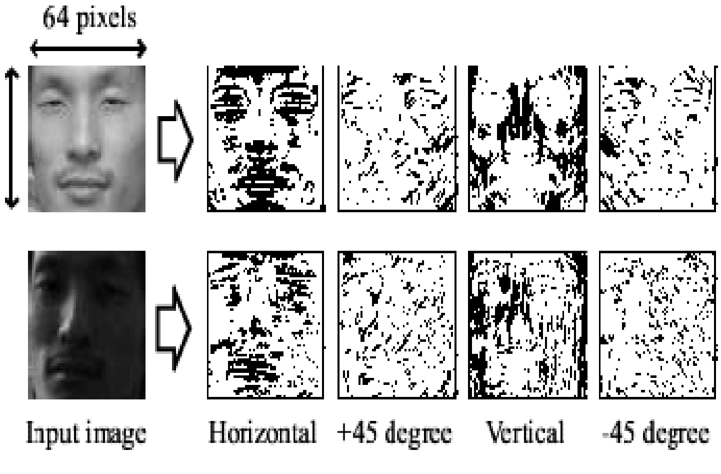


Fig 2 Feature map generated from dark and bright images

2.4.2 Feature Vectors

64-dimension feature vectors are generated from feature maps by taking the spatial distribution histograms of edge flags. In this work, three types of feature vectors, two general-purpose vectors and a face-specific vector generated from the same set of feature maps were employed to perform multiple-clue face detection algorithm [28]. Fig. 3 illustrates the feature vector generation procedure in the projected principal-edge distribution (PPED) [29]. This provides a general purpose vector. In the horizontal edge map, for example, edge flags in every four rows are accumulated and the spatial distribution of edge flags along the vertical axis is represented by a histogram. Similar histograms are generated from three other feature maps, and a 64- dimension vector is formed by concatenating the four histograms. Generation of the other general-purpose vector called the cell edge distribution (CED) vector is illustrated in Fig. 4. Each feature map is divided into 4x4 cells. Each element in a CED vector indicates the number of edge flags in the corresponding cell. A face specific feature vector generation scheme called the eyes-and-mouth (EM) extraction is shown in Fig. 5. Two16-pixel-high bands of rows are cut from the horizontal feature map. They correspond to the location of eyes and a mouth when the 64x64 pixel window encloses a human face. Then the number of edge flags in two neighboring columns is counted to yield a single vector element in a 64-dimension EM vector.

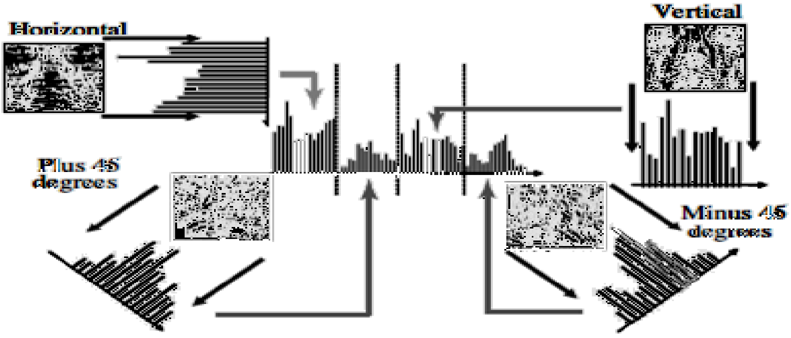


Fig3. Feature vector generation based on projected principal edge distribution.

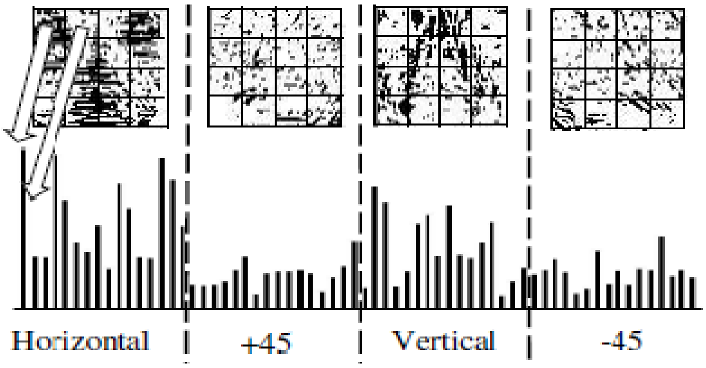


Fig. 4 Feature vector generation based on cell edge distribution algorithm.

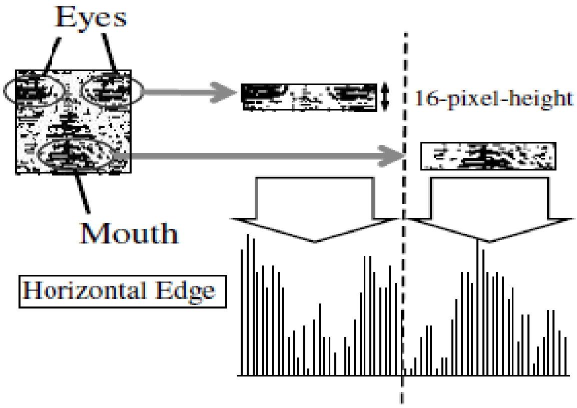


Fig.5 Feature vector generated by the eyes and mouth (EM) extraction.

2.4.3 Face Detection by Template Matching

Face detection is carried out in two steps: the coarse selection step and the fine selection step. In the coarse selection step, a 64x64-pixel area is taken from the input image and a feature vector is generated. Then, the feature vector is matched with all template vectors of face samples and non-face samples stored in the system and classified as a face or a non-face according to the category of the best-matched template vector. The matching is carried out using the Manhattan distance as the dissimilarity measure. Template matching is performed using three feature vector generation algorithms: PPED, CED and EM. If a local image is classified as a face by all the three vector representations, then it is adopted as a face candidate. This classification is carried out by pixel-by-pixel scanning of the 64x64-pixel window over the entire image.

2.4.4 Artificial Neural Network based Face Detection

In the field of face detection, as well as in the field of face recognition, a lot of literature is devoted to describing systems using Artificial Neural Networks (ANNs) as classifier. In the case of face recognition ANNs are used to determine whether a given face is one the system has seen before (and thus recognizes) despite differences in orientation, lighting and facial expressions of the same face. Therefore, the face that is offered to the system will be matched against a database of faces, and be classified as known or not known. In the field of face detection, however, ANNs are used to a certain representational model of the visual structure face.

In this approach neural networks are trained with a training set of face Images. The test image is then fed into the neural network for faces to be detected. In practice the network will not receive a perfect image of face which represented by vector as input. Specifically, the network should make as few mistakes as possible when classifying images with noise of mean 0 and standard deviation of 0.2 or less [17, 30]. Advantage of using a NN is that the system can be easily be upgraded to recognize faces.

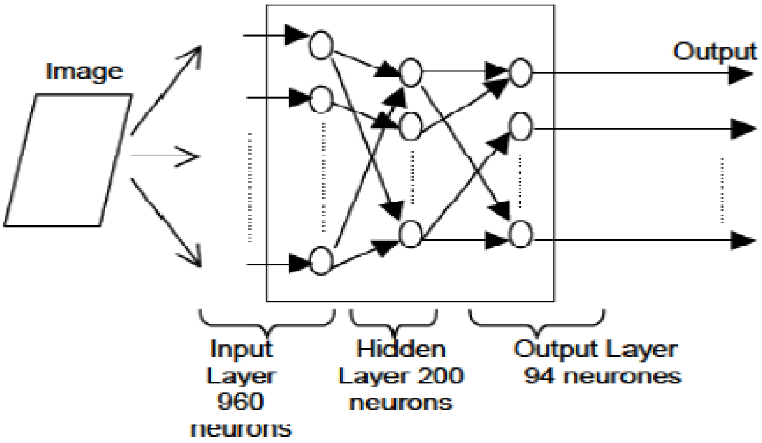


Fig.6 Neural Network

A different approach for detecting upright faces in grayscale images is proposed by Rowley et al. [31]. They propose to train a set of ANNs directly on the face and non-face input patterns. The ANNs take as input the information passed through by a window of receptive fields. This is a 20 x 20 pixel window which is divided into 26 sub-windows, each connected to one input unit of the network, thus reducing the number of input values from 400 to 26. In each experiment 2 or 3 networks are trained and during classification, arbitration takes place between the outputs of the different networks, by looking at the intersection or unification of the outputs of the different networks, or by voting between the outputs of the different networks.

2.4.5 Face detection based on prior knowledge

To build a model of the human face based on prior knowledge, one has to define features to distinguish between non-faces and faces. To do this we will look at the knowledge we humans possess about a face. A human being might define a face using knowledge of the appearance and the position of the facial features, such as the eyes, nose and mouth, and using knowledge about the context, such as hair, ears and neck. A face would thus be defined as:

A region of the human body between the **ears** and above the **neck**, possibly with **hair** on top or to the sides, containing **two eyes** on the same vertical position and a **nose** slightly lower and between the eyes, and a **mouth** even lower, stretching approximately the same distance to the left and right starting from the point right under the nose.

2.4.6 Skin color based Face Detection

For detecting face there are various skin color based algorithms have been proposed. Color is an important feature of human faces. Using skin-color as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features. Under certain lighting conditions, color is orientation invariant [32].However, color is not a physical phenomenon; it is a perceptual phenomenon that is related to the spectral characteristics of electromagnetic radiation in the visible wavelengths striking the retina. Tracking human faces using color as a feature has several problems like the color representation of a face obtained by a camera is influenced by many factors (ambient light, object movement, etc.), different cameras produce significantly different color values even for the same person under the same lighting conditions and skin color differs from person to person[33]. It is also robust towards changes in orientation and scaling and can tolerate occlusion well. A disadvantage of the color cue is its sensitivity to illumination Color changes and, especially in the case of RGB, sensitivity to illumination intensity. It would be fair to say that the most popular algorithm to face localization is the use of color information, whereby estimating areas with skin color is often the first vital step of such strategy. Hence, skin color classification has become an important task. Much of the research in skin color based face localization and detection is based on RGB, YCbCr and HSI color spaces. YCbCr and HSI are two most commonly used colour models for digital processing. In our work we have used the HIS model. Kjeldson and Kender defined a color predicate in HSV color space to separate skin regions from background [32]. Here the responsible values are hue (H) and saturation (S). If the threshold is chosen as [*H1, S1*] and [*H2, S2*], and a pixel is classified to have skin tone if the values [*H,S*] fall within the threshold and this distribution gives the localized skin image. Similar to above algorithm our approach is also having the same constraint.

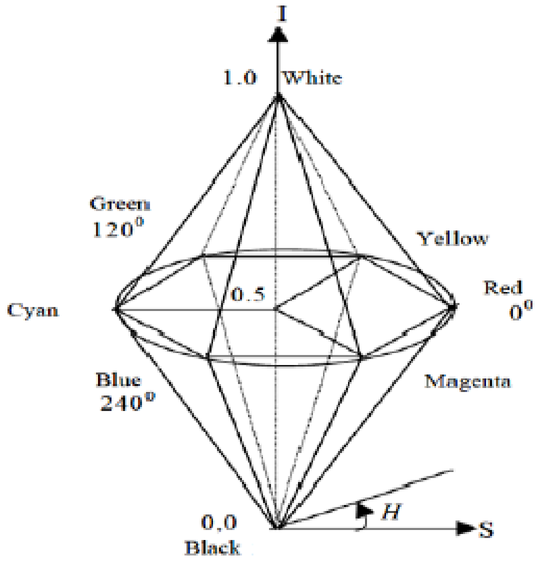


Fig.7 Double cone model of HIS color space

Kovac *et al.*, [34] has constructed classification rules which make it possible to classify skin color in images in different lighting conditions using the RGB color space. The classification takes two different lighting conditions into account, which are: uniform daylight and flash or lateral illumination. The algorithm was initially developed to work best under standard daylight illumination. The rules for classifying a pixel as skin are shown in Table1

Tab.1 Rules for skin pixel

Lighting condition	Uniform daylight	Flash or Lateral daylight
Skin classification rule	R>95,G>40,B>20 Max{R,G,B}-Min{R,G,B}>15 |R-G|>15,R>G,R>B	R>20,G>210,B>170 |R-G|<=15,B<R,B<G

Figure 8 shows the result of using the skin classification rules from Table on a test image.

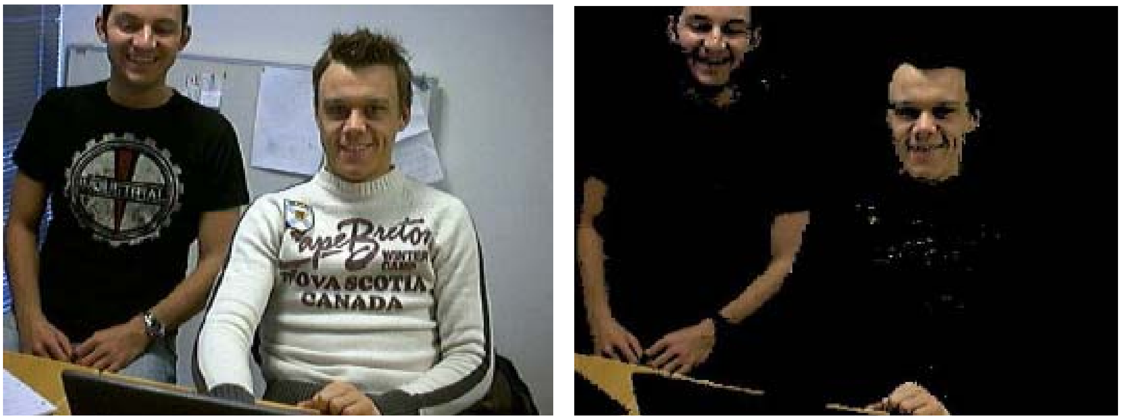


Fig.8 The skin segmentation rules from [Kovac et al., 2003] applied to an image.

In the test image used, the skin segmentation is performed very well. Almost all skin has been classified correctly and only the table in the image is wrongly classified as skin.

2.4.7 Texture Based Face Detection

Different researchers have published texture based face detection methods, based on training appearance models using the AdaBoost algorithm and using Haar-like features

[35 36 37] and Viola & Jones,et al. [38] . This section only describes these texture based approaches. The Haar-like features and the training approach have also been used in other applications. The basic idea was first proposed in [35] by Viola & Jones where two key ideas are presented. The first key idea is to create a face detector based on a combination of weak classifiers. To combine weak classifiers to a boosted classifier; a training algorithm based on AdaBoost is used. The second key idea is to create a cascade of boosted classifiers. A weak classifier is constructed of a single Haar-like feature, a boosted classifier is a weighted combination of weak classifiers, and a cascaded classifier is a sequence of boosted classifiers. An illustration of examples of the different types of classifiers and how they relate is shown in Figure 9.

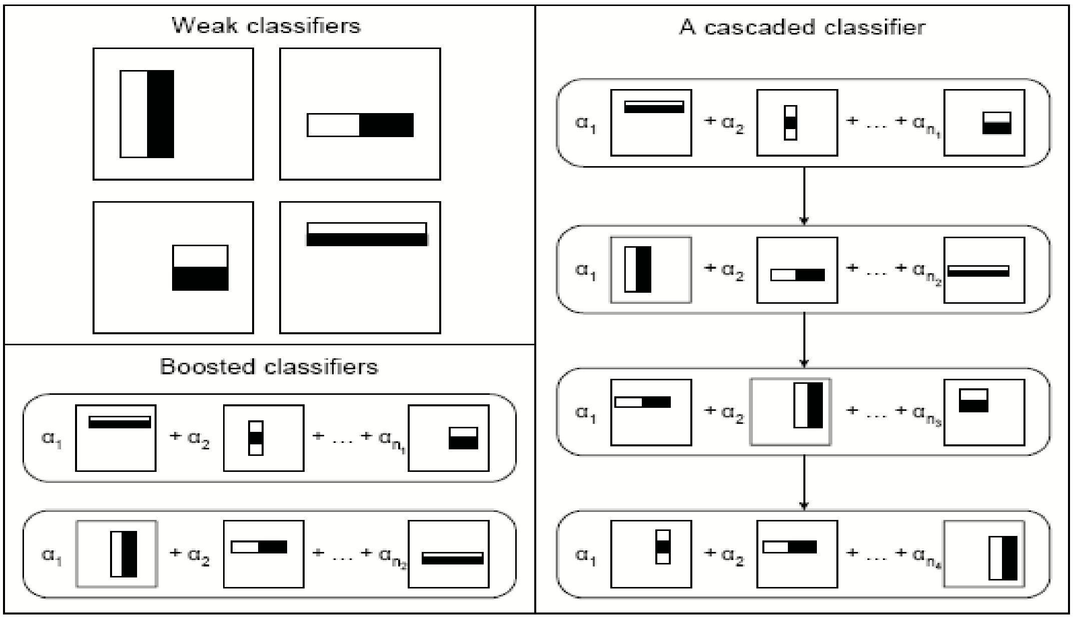

Weak classifiers
Boosted classifiers
a_1 + a_2 + ... + a_{n_1}
a_1 + a_2 + ... + a_{n_2}
A cascaded classifier
a_1 + a_2 + ... + a_{n_1}
a_1 + a_2 + ... + a_{n_2}
a_1 + a_2 + ... + a_{n_3}
a_1 + a_2 + ... + a_{n_4}

Fig.9 Example weak classifier, Boosted and cascade classifier

To detect faces in images a sub-window is moved across the image in different scales. Each sub-window is then processed by the cascaded classifier. Each boosted classifier in the cascade is denoted a layer, for each of these the sub-window is evaluated by the corresponding boosted classifier. If the sub window is classified as a face, it is passed to the next layer. A sub-window must be classified as a face by all layers of the cascade to be detected. This structure is illustrated in Figure 2.20. A cascaded classifier is trained to consist of increasingly more complex boosted classifiers. Each boosted classifier has a very high detection rate and a moderate false acceptance rate. This enables the first few layers of the cascaded classifier to reject a majority of the non face sub-windows in the input image, this way more computation time is spent on more difficult samples. The cascaded classifier is a fast detection algorithm, in [35] a frame rate of 15 frames per second is achieved while scanning an image of resolution 384x288, using a Pentium III 700MHz. However, the fast detection time is at the cost of training time.

Chapter 3

COLOR AND COLOR MODELS

Before we propose skin color model, it is useful to take a closer look at color in general. Because the way color is modeled in computers, is derived from the way the human eye perceives it, I will first describe the way the human eye processes color information.

3.1 Color in General and Human Color Perception

For an object to have color, it has to reflect visible light. Visible light is a small part of the electromagnetic spectrum, i.e. the part between 380 and 780 nm. If an object reflects or emits electromagnetic radiation between these two wavelengths, that object will be visible, and will be said to have color. For human beings to perceive this reflected or emitted light, it has to be received by the receptor cells that are situated in the human retina, which will pass this information to the visual cortex in the brain. Two sorts of receptor cells reside in the human eye: rods and cones. The rods are sensitive to light intensity only, so we don't use rods for perceiving color, but we do depend on them for perception in the dark. The cones are sensitive to wavelengths of light, so those are the ones we use for perceiving colors. The human retina contains three types of cones, all sensitive to a different part of the visible light spectrum. The first type is most sensitive to ±620 nm, the second type to ±520 nm and the third type to ±450 nm (respectively Red, Green and Blue). This means that human color perception is based on a combination of roughly these three colors.

3.2 The RGB Color Space

This system of Red, Green and Blue components is adopted by the designers of digital cameras and color monitors. A color camera, for instance, has sensitive cells for Red, Green a Blue intensities. These sensitive cells each receive an spectral energy distribution *E(λ)*, where **λ** are the wavelengths of the perceived light. Using these sensitive cells, the wavelengths of perceived light are mapped to RGB values using the

Following equations:

$$R = \int_{\lambda} E(\lambda) \cup_{R} (\lambda)\, d\lambda \qquad (\mathbf{3})$$

$$G = \int_{\lambda} E(\lambda) \cup_{G} (\lambda)\, d\lambda \qquad (\mathbf{4})$$

$$B = \int_{\lambda} E(\lambda) \cup_{B} (\lambda)\, d\lambda \qquad (\mathbf{5})$$

Where U_x | X €(R, G. B), are color component matching functions. The R, G and B components combined can produce every possible visible color. Hence we call RGB a complete color system. The RGB color space can be represented by a cube having R, G and B as its axes:

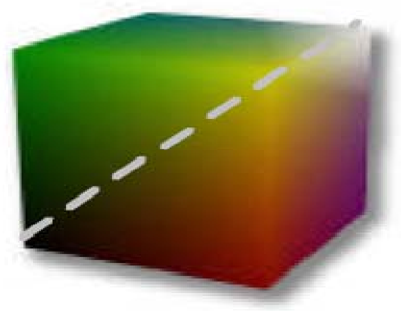


Fig.10 RGB cube with intensity diagonal

3.3 Intensity Invariant Color Spaces

The intensity of an image in RGB color space is defined by adding up the R, G and B components, i.e.:

$$I(R, G, B) = R+G+B \quad \textbf{(6)}$$

Applying this calculation to every pixel in an image yields an intensity or gray value image, which is devoid of all color information. From this we can derive that for acquiring *actual* color information we have no need for intensity information. In this subsection I will discuss some *intensity invariant* color spaces.

3.4 Hue Saturation Intensity (HSV, HSI)

Hue-saturation based colorspaces were introduced when there was a need for the user to specify color properties numerically. They describe color with intuitive values, based on the artist's idea of tint, saturation and tone. *Hue* defines the dominant color (such as red, green, purple and yellow) of an area; *saturation* measures the colorfulness of an area in proportion to its brightness [39]. The"intensity","lightness" or "value" is related to the color luminance. Several interesting properties of Hue were noted in [40]: it is invariant to highlights at white light sources, and also, for matte surfaces, to ambient light and surface orientation relative to the light source. However, [39], points out several undesirable features of these colorspaces, including hue discontinuities and the computation of "brightness" (lightness, value), which conflicts badly with the properties of color vision. .The transformation of RGB to HSV is invariant to high intensity at white lights, ambient light and surface orientations relative to the light source and hence, can form a very good choice for skin detection methods. The HSV color space is quite similar to the way in which humans perceive color. The other models, except for HSL, define color in relation to the primary colors. The colors used in HSV can be clearly defined by human perception, which is not always the case with RGB or CMYK.

$$\mathrm{H} = \arccos \frac{\frac{1}{2}((\mathrm{R}-\mathrm{G})+(\mathrm{R}-\mathrm{B}))}{\sqrt{((\mathrm{R}-\mathrm{G})^2+(\mathrm{R}-\mathrm{B})(\mathrm{G}-\mathrm{B}))}} \quad \textbf{(7)}$$

$$S = 1 - 3\frac{\min\{R,G,B\}}{(R+G+B)} \qquad \mathbf{(8)}$$

$$V = \frac{1}{3}(R + G + B) \qquad \mathbf{(9)}$$

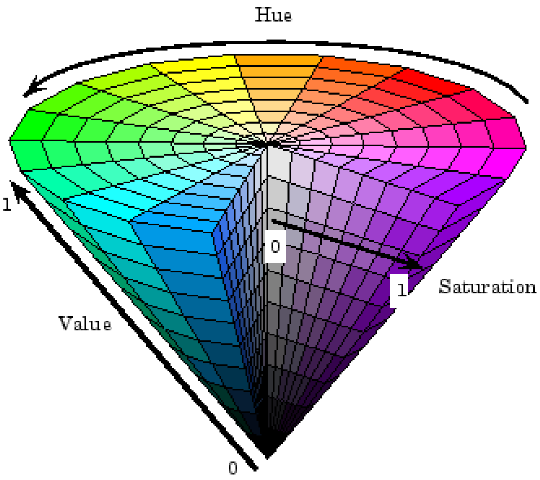


Fig.11 HSV color cone variation

An alternative way of hue and saturation computation using log opponent values was introduced in [40], where additional logarithmic transformation of RGB values aimed to reduce the dependence of chrominance on the illumination level. The polar coordinate system of Hue-Saturation spaces, resulting in cyclic nature of the colorspace makes it inconvenient for parametric skin color models that need tight cluster of skin colors for best performance. A different representation of Hue-Saturation using Cartesian coordinates can be used [41]:

$$X = S\cos H \qquad Y = S\sin H \qquad \mathbf{(10)}$$

3.5 TSL - Tint, Saturation, Lightness

A normalized chrominance-luminance TSL space is a transformation of the normalized RGB into more intuitive values, close to hue and saturation in their meaning.

$$S = \sqrt{\frac{9}{5}(r'^2 + g'^2)} \qquad \mathbf{(11)}$$

$$T = \begin{cases} \frac{\arctan\left(\frac{\mathrm{r}'}{\mathrm{g}'}\right)}{2\pi} + \frac{1}{4} & ,\mathrm{g}' > 0 \\ \frac{\arctan(\frac{\mathrm{r}'}{\mathrm{g}'})}{2\pi} + \frac{3}{4} & ,\mathrm{g}' > 0 \\ 0 \ , & \mathrm{g}' = 0 \end{cases} \qquad \mathbf{(12)}$$

$$\mathrm{L} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B} \qquad \mathbf{(13)}$$

Where r0 = r−1/3, g0 = g−1/3 and r, g come from (15). [41] Have compared nine different colorspaces for skin modeling with a unimodal Gaussian joint pdf (only chrominance components of the colorspaces were used). They argue that normalized TSL space is superior to other colorspaces for this task. [42] Has also employed this representation for their approach.

3.6 YCrCb

YCrCb is an encoded nonlinear RGB signal, commonly used by European television studios and for image compression work. Color is represented by *luma* (which is luminance, computed from nonlinear RGB [43]), constructed as a weighted sum of the RGB values, and two color difference values *Cr* and *Cb* that are formed by subtracting luma from RGB red and blue components.

$$\begin{aligned} \mathrm{Y} &= 0.299\mathrm{R}+0.587\mathrm{G}+0.114\mathrm{B} \\ \mathrm{Cr} &= \mathrm{R}-\mathrm{Y} \\ \mathrm{Cb} &= \mathrm{B}-\mathrm{Y} \end{aligned} \qquad \mathbf{(14)}$$

The transformation simplicity and explicit separation of luminance and chrominance components makes this color space attractive for skin color modeling [44].

3.7 The rgb Color Space

Another intensity invariant color space is the *rgb* color space. We have already stated that intensity is defined as the sum of the R, G, and B components. So another way to normalize with respect to intensity, is dividing this sum. Hence, the *rgb* color space is defined, with respect to the RGB space, as follows:

$$\mathrm{r} = \frac{\mathrm{R}}{\mathrm{R+G+B}}\ \mathrm{g} = \frac{\mathrm{G}}{\mathrm{R+G+B}}\ \mathrm{b} = \frac{\mathrm{B}}{\mathrm{R+G+B}} \qquad (15)$$

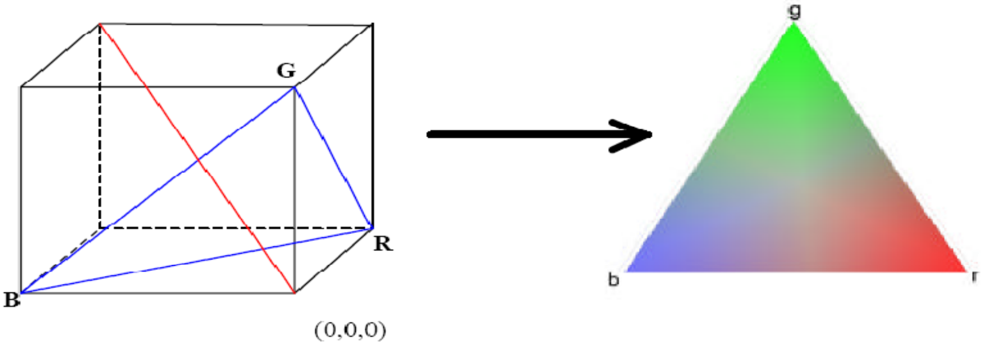


Fig.12 rgb normalized color space

As can be seen in figure 11, the R+G+B =1 plane in RGB color space defines an equilateral triangle. This triangle is called the chromatic triangle. The *rgb* color space is defined as the projection of points inside the RGB cube on the chromaticy cube. Since the intensity line makes a 90 degree angle with the chromaticy triangle, the colors that vary only in intensity, are projected on the same point on the chromatic plane, and consequently have the same *r*, *g* and *b* values. Thus, the *rgb* color space is intensity invariant. The *rgb* color space has multiple advantages. Since $r + g + b = 1$, one only needs two of these components to characterize all existing colors, thereby reducing the color space complexity. Furthermore the *rgb* color space is invariant to shadows, surface orientation, illumination direction and illumination intensity for dull matte surfaces, like the HSI color space. However, the system is unstable when intensity is low, and it is sensitive to illumination color. The *rgb* color space is not defined for achromatic color.

3.8 The c1c2c3 Color Space

Gevers et al. [45] defines the *c1c2c3* color space. This color space is defined as follows:

$$C1 = \arctan\left(\frac{R}{\max(G,B)}\right) \quad \mathbf{(16)}$$

$$C2 = \arctan\left(\frac{G}{max(R,B)}\right) \quad \mathbf{(17)}$$

$$C3 = \arctan\left(\frac{B}{max(R,G)}\right) \quad \mathbf{(18)}$$

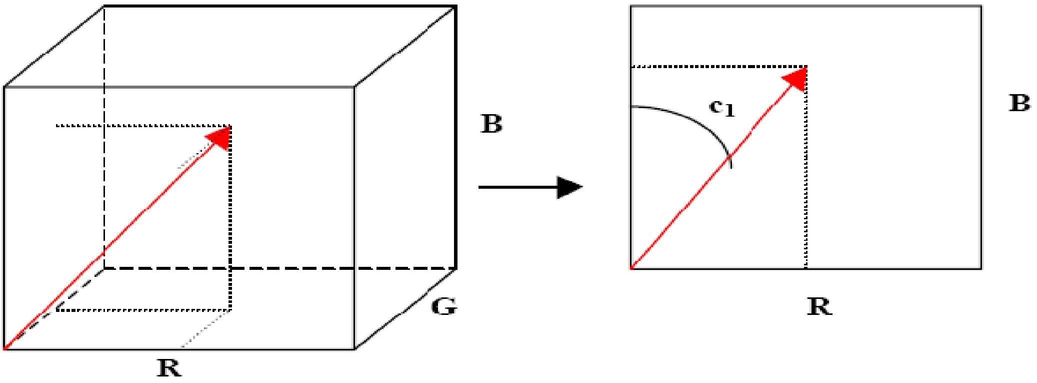


Fig.13 c1c2c3 color space

These values represent the angles of the body reflection vector, i.e. c1 is the angle between the red component and the maximum of the green and blue components. Figure 12 shows the calculation of c1 when the blue component has a greater value than the green component. This color space is invariant to shadows, surface orientation, and illumination direction and illumination intensity for dull matte surfaces.

Chapter 4

BACTERIA FORAGING

A new evolutionary technique, called Bacterial Foraging scheme, was introduced by K.M.Passino [46].The foraging can be modeled as an optimization process where bacteria seek to maximize the energy obtained per unit time spent during foraging. In this process, the nutrient function is defined and is being maximized by each bacterium in search of food. Each bacterium tries to maximize the amount of nutrient while minimizing time and energy cost by following four stages: 1) Chemo taxis, 2) Swarming, 3) Reproduction, and 4) Elimination & Dispersal. In the beginning, there will be as many solutions as the number of bacteria. So, each bacterium produces a solution for set of optimal values of parameters iteratively, and gradually all the bacteria converge on the global optimum.

During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an *E.coli* bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the above-mentioned algorithm the bacteria undergoes chemotexis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment. Figure 1 depicts how clockwise and counter clockwise movement of a bacterium take place in a nutrient solution.

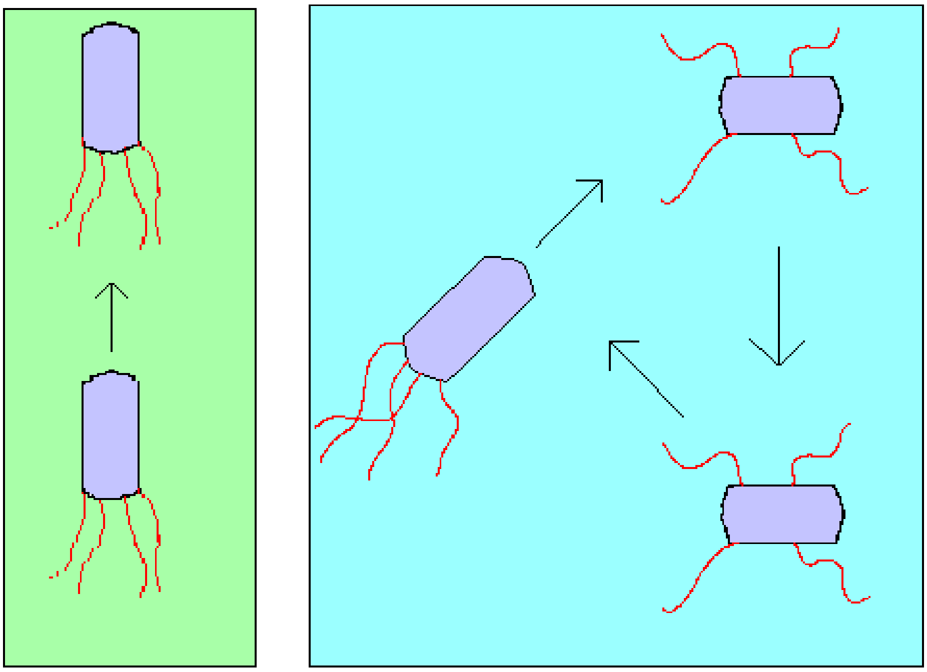


Fig.14. Swim and tumble of a bacterium

When they get food in sufficient, they are increased in length and in presence of suitable temperature they break in the middle to from an exact replica of itself. This phenomenon inspired Passino to introduce an event of reproduction in BFOA. Due to the occurrence of sudden environmental changes or attack, the chemotactic progress may be destroyed and a group of bacteria may move to some other places or some other may be introduced in the swarm of concern. This constitutes the event of elimination-dispersal in the real bacterial population, where all the bacteria in a region are killed or a group is dispersed into a new part of the environment.

Now suppose that we want to find the minimum of $J(\theta)$where $\theta\ p \in \Re$ (i.e. θ is a *p*-dimensional vector of real numbers), and we do not have measurements or an analytical description of the gradient $\nabla J(\theta)$. BFOA mimics the four principal mechanisms observed in a real bacterial system: chemotexis, swarming, reproduction, and elimination-dispersal to solve this non-gradient optimization problem. Let us define a chemotactic step to be a tumble followed by a tumble or a tumble followed by a run. Let *j* be the index for the chemotactic step. Let *k* be the index for the reproduction step. Let *l* be the index of the elimination-dispersal event. Also let

p: Dimension of the search space,

S: Total number of bacteria in the population,

N_c : The number of chemotactic steps,

N_s: The swimming length.

N_{re} : The number of reproduction steps,

N_{ed} : The number of elimination-dispersal events,

P_{ed} : Elimination-dispersal probability,

C (i): The size of the step taken in the random direction specified by the tumble.

Let $P(j, k, l)$ $\{ (j, k, l) \mid i$ $1,2,..., S\}$ $i = \theta =$ represent the position of each member in the population of the *S* bacteria at the *j*-th chemotactic step, *k*-th reproduction step, and *l*-th elimination-dispersal event. Here, let J (i, j, k, l) denote the cost at the location of the *i*-th bacterium $\theta_p^i(j, k, l)\in\Re$ (sometimes we drop the indices and refer to the *i*-th bacterium position as θ_ι). Note that we will interchangeably refer to *J* as being a “cost” (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, *S* can be very large (e.g., *S* =109), but $p = 3$. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. BFOA, however, allows $p > 3$ so that we can apply the method to higher dimensional optimization problems. Below we briefly describe the four prime steps in BFOA.

i) **Chemotaxis**: This process simulates the movement of an *E.coli* cell through swimming and tumbling via flagella. Biologically an *E.coli* bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose θ(i, j, k, l) represents i-th bacterium at jth chemotactic, k-th reproductive and l-th elimination-dispersal step. *C*(i) is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad (\mathbf{19})$$

Where Δ indicates a vector in the random direction whose elements lie in [-1, 1].

ii) **Swarming**: An interesting group behavior has been observed for several motile species of bacteria including *E.coli* and *S. typhimurium*, where intricate and stable spatio-temporal patterns (swarms) are formed in semisolid nutrient medium. A group of *E.coli* cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of *succinate*, release an attractant *aspertate*, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signaling in *E. coli* swarm may be represented by the following function.

$$J_{cc}(\theta, P(j,k,l)) = \sum_{i=1}^{S} J_{cc}\left(\theta, \theta^{i}(j,k,l)\right)$$

$$= \sum_{i=1}^{S}\left[-d_{attractant}\exp\left(-w_{attractant}\sum_{m=1}^{P}\left(\theta_m - \theta_m^i\right)^2\right)\right]$$

$$+\sum_{i=1}^{S}\left[-h_{repellant}\exp\left(-w_{repellant}\sum_{m=1}^{P}\left(\theta_m - \theta_m^i\right)^2\right)\right] \qquad \textbf{(20)}$$

where $J\ cc$ (θ, P (j, k, l)) is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, S is the total number of bacteria, p is the number of variables to be optimized, which are present in each bacterium and $\theta = [\theta_1, \theta_2, \dots, \theta_P]^T$ is a point in the p-dimensional search domain. $d_{attractant}, w_{attractant}, h_{repellant}, w_{repellant}$ are different coefficients that should be chosen properly.

iii) **Reproduction:** The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

iv) **Elimination and Dispersal**: Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

The pseudo-code of the complete algorithm is presented below:

4.1 The BFOA Algorithm

Parameters:

[Step 1] Initialize parameters *p, S*, N_c, N_s, N_{re}, N_{ed}, P_{ed}, *C(i)(i=1,2…S)*, θ^i.

Algorithm:

[Step 2] Elimination-dispersal loop: *l*=*l*+1

[Step 3] Reproduction loop: *k*=*k*+1

[Step 4] Chemotaxis loop: *j*=*j*+1

[a] For *i* =1,2…S take a chemotactic step for bacterium *i* as follows.

[b] Compute fitness function, *J (i, j, k, l).*

Let, $J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l), P(j,k,l))$ (i.e. add on the cell-to cell Attractant–repellant profile to simulate the swarming behavior)

Where, J_{cc} is defined in (2).

[c] Let J_{last}=*J (i, j, k, l)* to save this value since we may find a better cost via a run.

[d] Tumble: generate a random vector Δ(*i*)∈$\Re p$ with each element (*i*),*m* 1,2,..., *p*, *m* Δ = a random number on [-1, 1].

[e] Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad \textbf{(21)}$$

This results in a step of size *C*(*i*) in the direction of the tumble for bacterium *i.*

[f] Compute *J* (*i*, *j* +1, *k*, *l*) and let

$$J(i,j+1,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l))$$

[g] Swim

i) Let *m*=0 (counter for swim length).

ii) While $m< N_s$ (if have not climbed down too long).

• Let $m=m+1$.

• If J $(i, j+1, k, l) < J_{last}$ (if doing better), let $J_{last} = J(i, j+1, k, l)$ and let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad \mathbf{(22)}$$

And use this $\theta^i(j+1,k,l)$ to compute the new J $(i, j+1, k, l)$ as we did in

[f] Else, let $m= N_s$. This is the end of the while statement.

[h] Go to next bacterium $(i+1)$ if $i \neq S$ (i.e., go to [b] to process the next bacterium).

[Step 5] If $j < N_c$, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

[**Step 6**] Reproduction:

[a] For the given k and l, and for each $i = 1,2,..., S$

$$J^i_{health} = \sum_{j=1}^{N_c+1} J(i,j,k,l) \qquad \mathbf{(23)}$$

Let be the health of the bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health} (higher cost means lower health).

[b] The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

[**Step 7**] If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

[**Step 8**] Elimination-dispersal: For $i = 1,2...,$ S with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise end.

In the chemo taxis stage, the bacteria either resort to a tumble followed by a tumble or make a tumble followed by a run or swim. This is the movement stage of bacteria through swimming and tumbling. On the other hand, in swarming, each *E. coli* bacterium signals another via attractants to swarm together. This is basically the cell to cell signaling stage. Furthermore, in reproduction bacterium with the least energy dies and the other bacteria with high energy survive. While in the elimination and dispersal stage, any bacterium from the total set can be either eliminated or dispersed to a random location during the optimization process. This stage helps the bacteria avoid the local optimum.

Chapter 5

PROPOSED APPROACH

The Face Detection in proposed approach is developed by detecting skin pixels first in a given image using the bacteria foraging. The proposed face detection method based on Bacteria foraging, an optimization technique. A number of Bacteria agents with evolutionary behaviors are uniformly distributed in the 2-D image environment to search the skin-like pixels and locate each face-like region by evaluating the local color distribution.

5.1 Skin Color Model for Face Detection

It is very important to select a color model that can detect face of different race and sex. The selected color space must be able to distinguish skin color region from the non-color region. There are many color space but certainly all are not suitable for skin color detection. Therefore color model should de chosen appropriately for the skin color detection. The basis of color space can be application and the effectiveness of that color space in application.

The color space including RGB, YCrCb [47], HSV [48] and CIELUV [49] are usually used for skin color detection. Some research results show [50] [48] that the HSV (hue, saturation, and value) model is more Similar to the human perception of color. In this paper we proposed to use the HSV color space model for skin like region. The reason for opting this color space is more compatible to human perception of color and the hue and saturation components have been reported also to be sufficient for discriminating color information for modeling skin color. Another reason for opting this color space that, this color model is invariant to the unknown lighting condition. From this point of view the model derived from RGB color will be inferior to HSV. The skin color detection area in [49] is defined in HSV color space as (27).

$$skin\,pixel = \begin{cases} 1 & 0.04 < H < 0.1;\ 0.129 < V < 0.6812;\ 0.2 < S < 0.729 \\ 0 & \end{cases} \quad (\mathbf{24})$$

5.2 Skin color Detection using Bacteria Foraging:

Using the bacteria foraging to extract the face from color image is a new method. In this paper we proposed a new method for detecting and segmentation of skin region and face detection in color image. Unlike the other method such as template matching, proposed method requires less computation time. In the Proposed approach, we have made certain assumption as: a pixel in the image represents a bacterium and nutrient function is based upon the presence of skin color.

The algorithm for the proposed approach proceeds as follows:

A. Chemotexis step:

The chemotexis step decides the movement of bacteria in which direction it will move. Depending upon the rotation of flagella each bacterium decides whether it should swim (move in a predefined direction) or tumble (move in an altogether different direction). In our approach bacteria search for the skin color pixel in the image. The movement of the bacterium is decided by the equation (27).

B. Reproduction Step:

With sufficient amount nutrient, the bacterium will split into more than one bacterium such that each bacterium will have sufficient health. In reproduction step if more than one position of the 8 neighbor of current position of bacterium satisfies the equation, then bacterium will split into number of bacteria equal to skin color pixel.

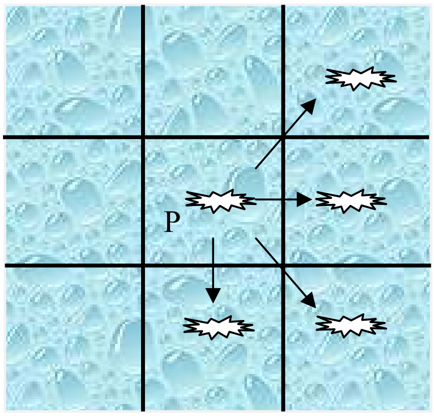


Fig.15 Bacterium reproduction

Let the p be current position of bacterium and four neighbor of it satisfies the equation (27) then current bacterium will split in the respective position.

C. Elimination and dispersal

If bacterium reaches a location with no or less amount of nutrient then it will be eliminated to a new location with more nutrient health. If all the 8 neighbor locations of current bacterium do not belong to skin like region, bacteria will eliminate and disperse to other location randomly

The algorithm using the bacteria foraging approach to detect a face like region in a color image can be described below

1. Randomly distribute the initial set of bacteria over the image containing skin like Region as show in fig. blue pixel represent a bacteria.
2. For each bacterium compute the HSV value of eight neighborhood pixel at the location Where a bacterium is present then determines whether that location represent a skin like pixel or not using the Equation(27).

3.If the location belong to skin like region and has not been visited by any other bacteria, then move the current bacteria towards next skin color pixel

4. Spilt bacteria equal to the connected skin color pixel in neighborhood of current Position.

5. If the location does not belong to skin like region, bacteria will eliminate and disperse to other location randomly

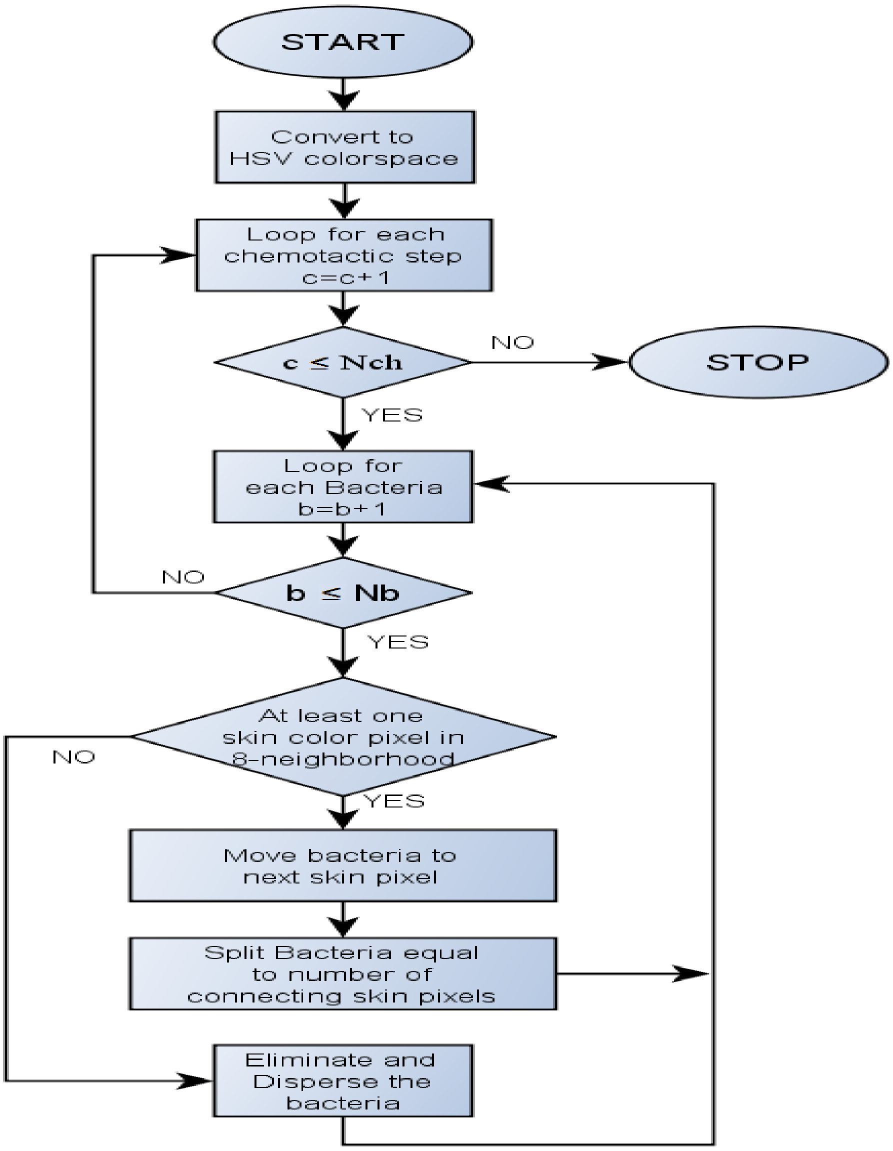

START
Convert to HSV colorspace
Loop for each chemotactic step c=c+1
c ≤ Nch
NO
STOP
YES
Loop for each Bacteria b=b+1
NO
b ≤ Nb
YES
At least one skin color pixel in 8-neighborhood
NO
YES
Move bacteria to next skin pixel
Split Bacteria equal to number of connecting skin pixels
Eliminate and Disperse the bacteria

Fig.16 Flow of proposed Algorithm

5.3 Face shape estimation

After face like region has been estimated we can estimate the face shape by utilizing the shape information. An approach to estimate the face is by finding the best fit ellipse. The best fit ellipse can be computed with the help of least square estimation method. General conic equation of an ellipse can be represented as,

$$G(x,y;\beta) = ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0 \qquad (\mathbf{25})$$

β is vector of unknown(a,b,c,d,f,g) parameter to be estimated

To find the values of parameters a, b, c, d, f and g, for best fit on the face shape least square estimation technique is used. To fit the best ellipse from given set of points $(x_{1,}y_1)$, (x_2,y_2), …, (x_n,y_n) n ≥ 5, error needs to be minimized as

$$error \ = \sum_{i=0}^{n} [r_i]^2 \qquad (\mathbf{26})$$

r_i is the residual which define as below

$$r_{i=}G_i - G_i(x,y;\beta) \qquad (\mathbf{27})$$

To obtain least square error, take partial derivative with respect to β

$$\frac{\partial g}{\partial \beta_i} = 2\sum_i r_i \frac{\partial r_i}{\partial \beta_i} = 0 \qquad (\mathbf{28})$$

Where j=1,2,3,…………………m and m is the no. of unknown parameter

Using the eq. (31) we can get the unknown parameter i.e. (a,b,c,d,f,g)

Now we can calculate center of ellipse as:

$$X_0 = \left(\frac{cd - bf}{b^2 - ac}\right) \qquad (\mathbf{29})$$

$$Y_0 = \left(\frac{af - bd}{b^2 - ac}\right) \qquad (\mathbf{30})$$

Chapter 6

EXPERIMENTAL RESULTS AND CONCLUSIONS

To evaluate the performance of our approach numerically, we randomly select 100 images from the FERET face skin database as the test data. The Color FERET database contains a total of 11338 facial images. They were collected by photographing subjects over the course of 15 sessions between 1993 and 1996. This database is largely a color version of the original Facial Recognition Technology (FERET) Database, which was released in 2001 and consisted of 14051 grayscale images of human heads with views ranging from frontal to left and right profiles. The 994 subjects (Color FERET database), 1209 subjects (Grey FERET database). Color depth 24-bit color images and eight-bit grayscales images.

A duplicate set is a second set of images of a person already in the database and was usually taken on a different day. For some individuals, over two years had elapsed between their first and last sittings, with some subjects being photographed multiple times. This time lapse was important because it enabled researchers to study, for the first time, changes in a subject's appearance that occur over a year.

Table 2 is the comparison of detection rate (DR) and false rejection (FRs) for the proposed methods, and Fig. 19 shows the detected faces by the proposed method. In Fig. 19, the ellipse indicates the location and size of the detected face. Fig. 17 shows result of skin color segmentation using proposed approach. It can be observed from the able 2; the proposed method has a better detection rate, and less false rejection errors.

Table 2 Evaluation of Performance Parameter

Approaches	**No. of faces**	**DR (%)**	**FR**
Proposed Method	100	94	6

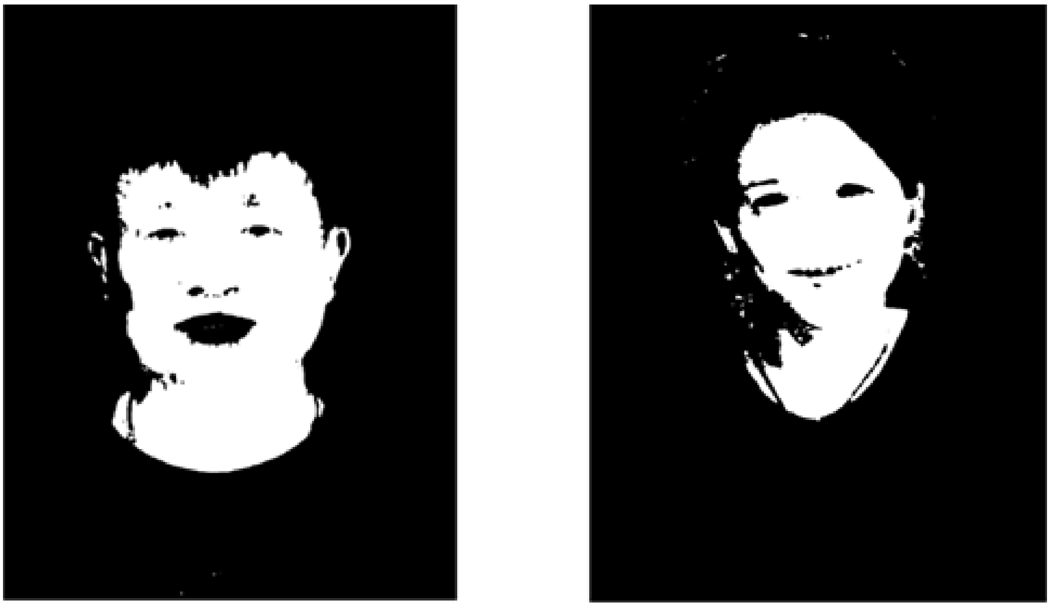


Fig.17 skin color segmentation using bacteria

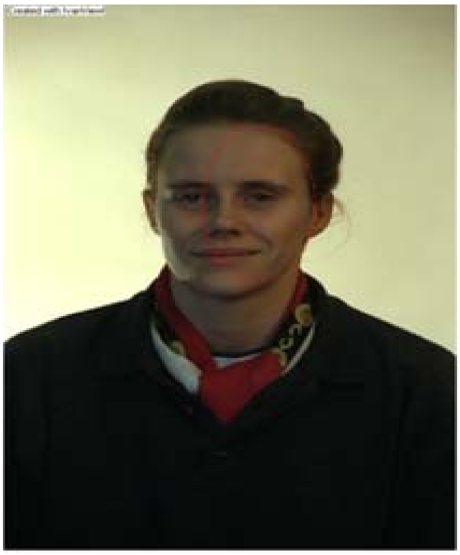
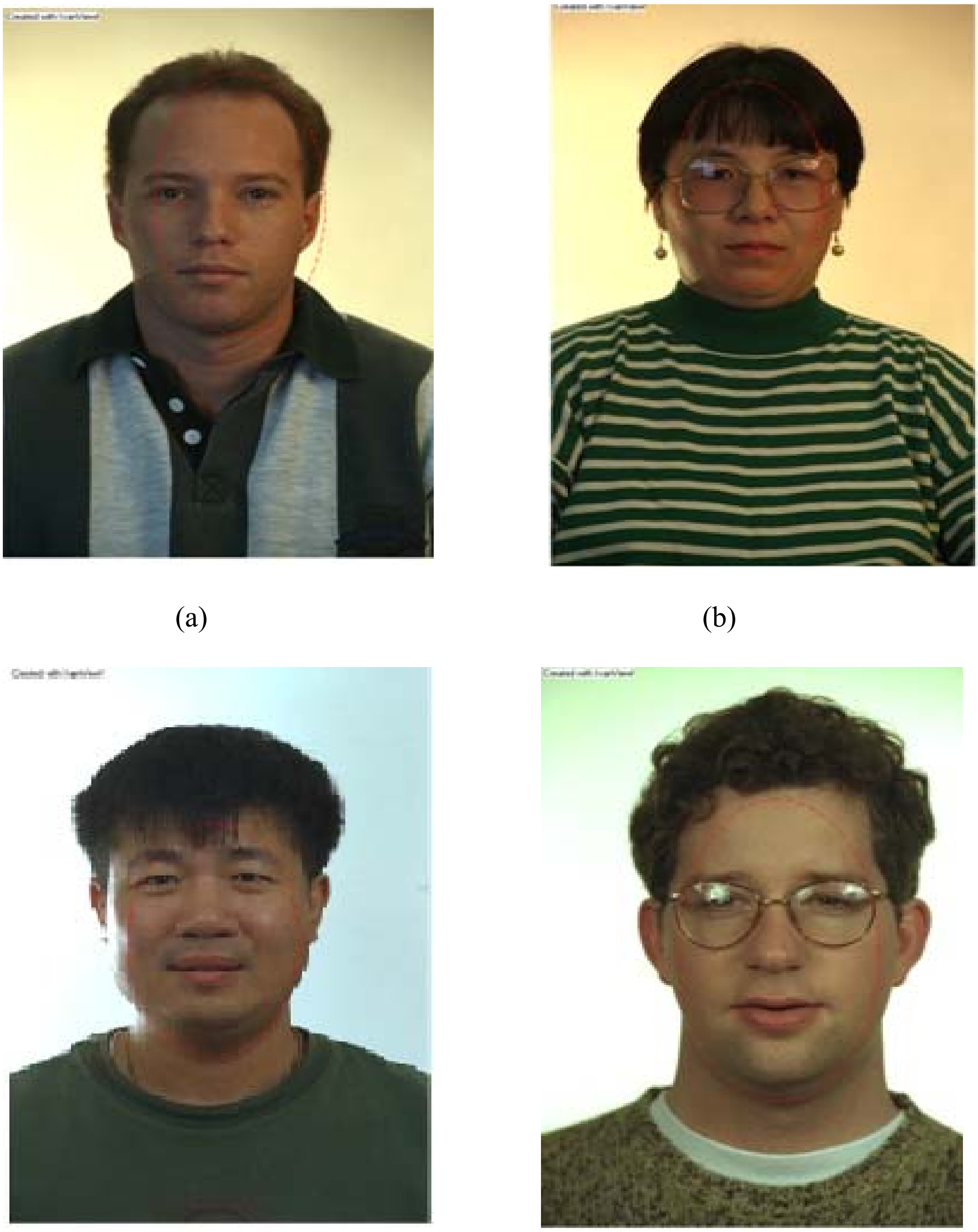


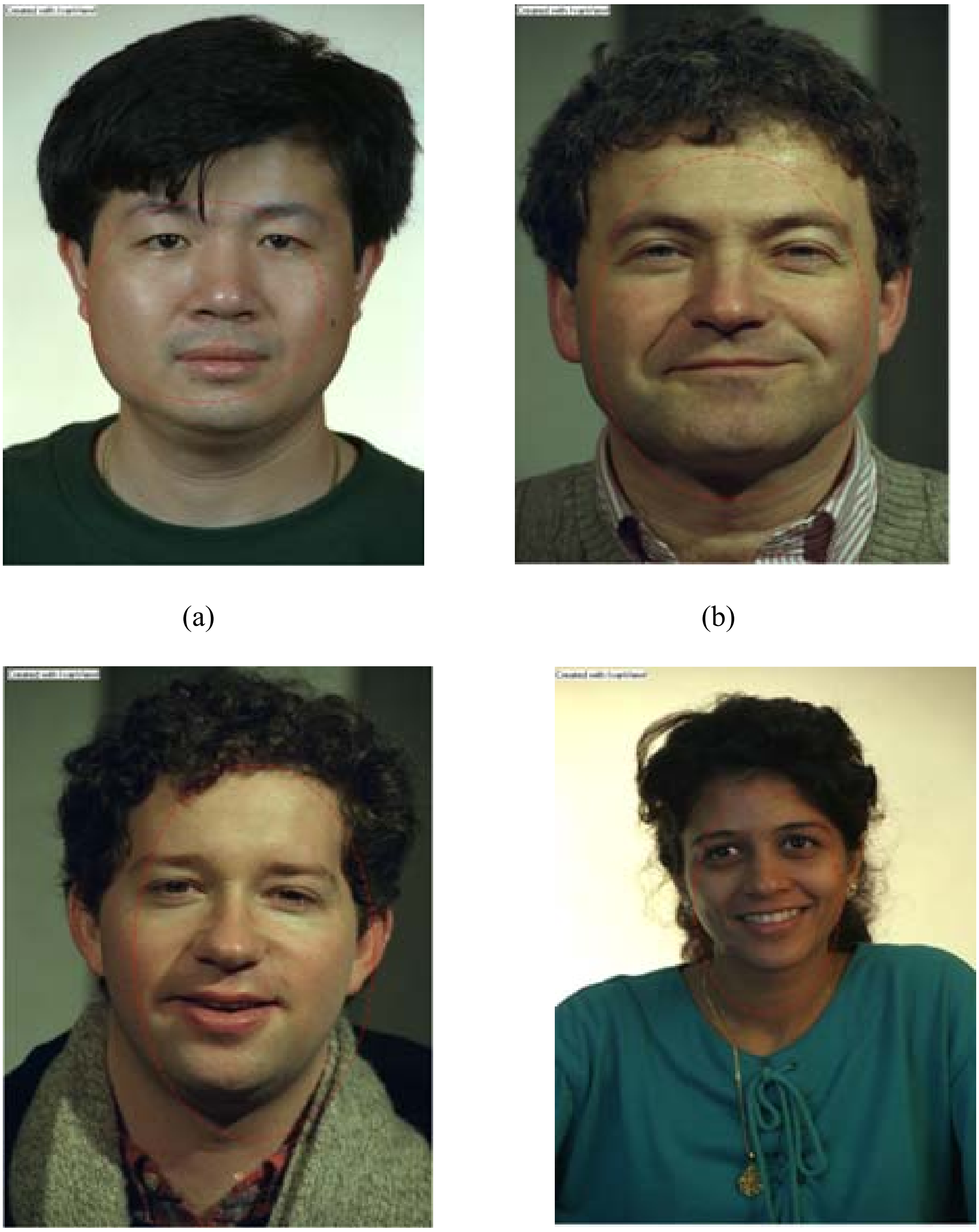
Fig.18 False Rejection



(a) (b)

(c) (d)

Fig. 19 Face Detection using proposed method



(a)

(b)

(c)

(d)

Fig. 20 Some more Result using Proposed Method

Fig 20 shows the pose variation face detection using the proposed method; therefore the proposed method is also able to detect faces in pose variation.

6.1 Conclusion

Human face perception is currently an active research area in the computer vision community. Face detection falls into the field of *computer vision and pattern recognition*. Face Detection is a very important step in recognition of faces. Face recognition is done mainly in two step face detection and then recognition. The recognition can only be possible if face is detected appropriately by face detection method used.

We have made an attempt to detect faces in colored images based using bacteria foraging technique. It has been employed in conjunction with skin color segmentation. As a result the occurrence of false negative has been greatly reduced. The work presented in this thesis report has been tested in matlab. The testing has been done using standard datasets [FERET] to compare our result with prevailing face detection techniques.

False detection is also low and this shows that it is able to distinguish between actual skin and background color. It is also seen that our algorithm is capable of classifying skin region in complex colored image. The robustness of the algorithm against the variance of illumination, focus and scales has been checked for a number of sample images.

References

1. Froba B. and Kublbeck C. “Robust Face Detection at Video Frame Rate Based on Edge Orientation Features”, Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition PP. 342-347,Washington DC,USA,21 May,2002.
2. Nara Y., and Yusuke ,“Face Detection Using the Shape of Face with Both Color and Edge” Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems Singapore, vol.1,PP. 147-152 December, 2004
3. Lee J.M., Kim J.H., and Moon Y.S. “Face Extraction Method Using Edge Orientation and Face Geometric Features” International Conference on Convergence Information Technology, PP. 1292-1297,21 Nov. IEEE 2007
4. Wu B. and nevatia R. “ Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors” International Journal of Computer Vision, Volume 75(2), PP. 247–266,2007
5. Liu J. And Huang H. “Study of Human Face Image Edge Detection based on DM642” VOl.-8, PP. 175-179, ICCSIT, 2010 IEEE
6. YX Wei-bo and Du Zun-sheng “An improved Kirsch human face image edge-detection method based on Canny algorithm”, PP 4740-4743 16 April IEEE 2011.
7. Shaick B. and Yaroslavsky L. “Fast Face Detection Using Hash Table of Large Set of Templates” Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, Vol.-2, PP 674-679, 10Sept, 2003.
8. J.W. Seo, M.S. Song, J.H. Park, D.U. Ahn, and S.J. Jhang, “Face Detection and Ellipse Modeling Based on Skin Color,” IEEK Conf., PP. 421, 1999

9. J.W. Seo, M.S. Song, J.H. Park, D.U. Ahn, and S.J. Jhang, "Face Detection and Ellipse Modeling Based on Skin Color," IEEK Conf., PP. 421-427, 1999

10. Wang J. and Tan T. "A new face detection method based on shape information" PATTERN RECOGNITION LETTERS volume-21, PP. 463-471

11. B. Moghaddam, and A. Pentland, "Probabilistic visual learning for object representation", IEEE Transaction on Pattern Analysis and Machine Intelligence volume 19 PP. 696–710 1997.

12. Hsu R.L., M. Abdel-Mottaleb, and Jain A.K., "Face detection in color images", IEEE Transaction on Pattern Analysis and Machine Intelligence volume 24 (5) PP 696–706 2002.

13. Tian Y., Kanade T., and Cohn J., "Robust lip tracking by combining shape color and motion", in: Proceeding of the 4th Asian Conference on Computer Vision (ACCV), , PP. 1040–1045. January 2000

14. Lee.T. and Park S.K, "An effective method for detecting facial features and face in human–robot interaction" INFORMATION SCIENCES vol.-176 PP 3166–3189 2006

15. Choong Yow K and Cipolla R. "Feature-based human face detection" IMAGE AND VISION COMPUTING volume 15 PP 713-735 ,1997

16. Ben-Yacoub S., Fasel B., and Lttin J. "Fast face detection using mlp and fft". In Audio And Video-Based Biometric Person Authentication, PP 31–36, Washington, D.C., 1999.

17. H.A.Rowley. "Neural Network-Based Face Detection". PhD thesis, Carnegie Mellon University, Pittsburgh, 1999.

18. B. Fasel.” Fast multi-scale face detection”. IDIAP-COM 4, IDIAP, 1998.

19. S.-H. Lin, S.-Y. Kung and L.-J. Lin. “Face recognition/detection by probabilistic decision based neural network”. IEEE Transactions on Neural Networks, Special Issue on Artificial Neural Networks and Pattern Recognition,Vol.8, PP. 114-132,1 January 1997

20. FERAUD R. and BERNIERA O. “Constrained Generative Mode Applied to Face detection” Neural letter processing vol.5 PP. 11-19, 1997.

21. Wu H., Chen Q., and Yachida M. “Face Detection From Color Images Using a Fuzzy Pattern Matching Method” Transactions on pattern analysis and machine intelligence, VOL. 21, PP 6, JUNE 1999, IEEE.

22. Yorita A. and kabota N. “Multi-stage fuzzy evaluation in evolutionary robot vision for face detection” Evolutionary intelligence volume 3, PP 67–78, Springer-Verlag 2010.

23. Kubota N, and Tomioka Y “Evolutionary robot vision for human tracking of partner robots in ambient intelligence”. In Proceedings (CD-ROM) of congress of evolutionary computation, PP 1491–1496, 2007.

24. Sulistijono IA., Kubota N.,“Human head tracking based on particle swarm optimization and genetic algorithm”. J Adv. Comput. Intell Intell Inform 11, No. 6, PP 681–687, 2007.

25. Chellappa R., Wilson C. L., and Sirohey S., “Human and machine recognition of faces: A survey”. Proceedings of the IEEE,83(5):705–740, 1995

26. Kondo T. and Yan H., “Automatic human face detection and recognition under on uniform illumination”, .Pattern Recognition Letters, vol. 32, PP.1707-1718, 1999.

27. Rowley H., Baluja S., and Kanade T. “Neural Network-Based Face Detection” Trans. Pattern Analysis and Machine Intelligence, Vol. 20, no. 1, pp. 23-38, Jan.1998, IEEE.

28. Suzuki Y. and Shibata T., “Multiple-Clue Face Detection Algorithm Using Edge-Based Feature Vectors”, Accepted for presentation in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2004, IEEE.

29. Yagi M. and Shibata T., “An Image Representation Algorithm Compatible with Neural-Associative-Processor-Based Hardware Recognition Systems”, Trans. Neural Networks, Vol. 14, no. 5, pp.1144-1161, Sep. 2003, IEEE.

30. Howell N. and Buxton H. “rbf network methods for face detection and attentional frames”, NEURAL PROCESSING LETTERS vol.15, pp 197–211, 2002.

31. Rowley, H.A., Baluja, S., Kanade, T.,. “Neural network based face detection.” Trans. Pattern Anal. Machine Intell.Vol. 20, pp 23–38., 1998, IEEE.

32. Kjeldsen, R. and Kender, J., “Finding Skin in Color Images,” Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, PP. 312-317, 1996.

33. Suzuki, Y. and Shibata, T., “an edge-based face detection algorithm robust against illumination, focuses, and scale variations”. European signal processing PP. 6-10 2004.

34. Kovac, J., Peer, P. and Solina, F. 2d versus 3d color space face detection. In EURASIP Conference on Video Image Processing and Multimedia Communications EC-VIP-MC’03 pp. 449–454. 2003

35. P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”. Computer Society Conference on Computer Vision and Pattern Recognition,vol.-1;pp 511-518 IEEE, 2001

36. Lienhart, R. and Maydt, J., “An extended set of haar-like features for rapid object detection.” ICIP 2002 vol. 1, pp. 900–9003. In IEEE, 2002.

37. Wu and J.Rehg and M. Mullin. “Learning a rare event detection cascade by direct feature selection.”,Advance in neural information processing System MIT Press.(2004)

38. Viola, P. and Jones, M. J. “robust real-time face detection.” Int. J. Comput. Vision,vol. 57 (2), pp 137–154. , (2004).

39. Skarbek W. and koschan A. “Color image segmentation – a survey”. Tech. report. Institute for Technical Informatics, Technical University of Berlin, 1994 October.

40. Brown, D., Craw, I., and Lewthwaite, J. “A som based approach to skin detection with application in real time systems”. In Proc. of the British Machine Vision Conference, 2001.

41. Fleck, M., Forsyth, D. A., AND Bregler, C., “Finding naked people”. In Proc. of the ECCV, vol. 2, 592–602. 1996.

42. Phung, S. L., Bouzerdoum, A., AND Chai, D. “A novel skin color model in ycbcr color space and its application to human face detection”. In International Conference on Image Processing (ICIP), vol. 1, 289–292., 2002 IEEE.

43. POYNTON, C. A. Frequently asked questions about color. In ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz.

44. Hsu, R.-L., Abdel-Mottaleb, M., AND Jain, A. K. "Face detectionin color images". IEEE Trans. Pattern Analysis and Machine Intelligence24, 5, 696–706., 2002.

45. Gevers,Th. and Smeulders A.W.M., "Color Based Object Recognition" , Pattern Recognition (PR), 32, pp. 453-464, March, 1999

46. Passino, K.M., Biomimicry of bacterial foraging for distributed optimization and control, Control Systems Magazine, IEEE, Jun 2002, Volume: 22, pp. 52-67

47. Kim, D. H. and Chien H. C. Do, S. II, "Preferred skin color reproduction based on adaptive affine transform", IEEE Trans. Consumer Electron., vol. 51, no. 1, pp. 191-197, Feb. 2005

48. You Y., Chien H. C. Do, S. II and Tae H. S., "Preferred skin color reproduction based on affine transform and contour reduction function using erosion". SID, pp. 1598-1601, 2006.

49. Yendrikhovskij S. N., Blommaert F. J. J., and H. D. Ridder, "Optimizing color reproduction of natural images", Proc. of Sixth Color Imaging. Conference: Color Science, Systems, and Applications, pp. 140-145, 1998.

50. Sobottka K. and Pitas I., "A Novel Method for Automatic Face Segmentation, Facial Feature Extraction and Tracking", Signal Processing: Image Communication. vol. 12, no.3, pp.263-28 Mar. 1998. S

51. Kim H.I., Lee S.H. and Cho N.I. "Rotation-invariant face detection using angular projections" ELECTRONICS LETTERS, Vol. 40 PP. 12 ;10th June 2004