Facial Expression Recognition for Human-Robot Interface

Mohammad Ibrahim Khan[†] **Md. Al-Amin Bhuiyan**[††]

[†]Dept. of Computer Science & Engineering, Chittagong University of Engineering and Technology, Chittagong-4349, Bangladesh.
[††] Dept. of Computer Science & Technology, Jahangirnagar University, Savar, Dhaka – 1342, Bangladesh.

Summary
This paper presents an approach to recognize human facial expressions for human-robot interaction. For this, the facial features, especially eyes and lip are extracted and approximated using Bézier curves representing the relationship between the motion of features and changes of expressions. For face detection, color segmentation based on the novel idea of fuzzy classification has been employed that manipulates ambiguity in colors. Experimental results demonstrate that this technique can robustly classify skin region and non-skin region. In order to decide whether the skin region is face or not, largest connectivity analysis has been employed. This method can recognize the facial expression category, as well as the degree of facial expression change. Finally, the system has been implemented by issuing facial expression commands to a manipulator robot.

Key words:
Bézier curves, Facial expression, Facial Action Coding System (FACS), Face detection, Fuzzy classification, Skin color, YCbCr..

1. Introduction

Facial expression analysis has been attracted considerable attention in the advancement of human-machine interface since it provides a natural and efficient way to communicate between humans. Some application areas related to face and its expressions include personal identification and access control, video phone and teleconferencing, forensic applications, human-computer interaction, automated surveillance, cosmetology, and so on. But the performance of the face detection certainly affects the performance of all the applications.

Many methods have been proposed to detect human face in images, they can be classified into four categories: knowledge-based methods, feature-based methods, template-based methods and appearance-based methods. When used separately, these methods cannot solve all the problems of face detection like pose, expression, orientation, occlusion. Hence it is better to operate with several successive or parallel methods.

Most of the facial expression recognition methods reported to date are focused on recognition of six primary expression categories such as: happiness, sadness, fear, anger, disgust and grief [1]. For a description of detailed facial expressions, the Facial Action Coding System (FACS) was designed by Ekman and Friensen in the mid 70's [2]. In FACS, motions of the muscles of the face are divided into 44 action units and any facial expression are described by their combinations. Synthesizing a facial image in model based image coding [3] and in MPEG-4 FAPs has important clues in FACS. Using MPEG-4 FAPs, different 3D face models can be animated. Moreover, MPEG-4 high level expression FAP allows animating various facial expression intensities. However, the inverse problem of extracting MPEG-4 low and high level FAPs from real images is much more problematic due to the fact that the face is a highly deformable object. Furthermore, when it comes to recognition up to facial expression intensities in real time, subtle failure in facial area segmentation condition cause crucial problems. For facial image synthesizing applications, many approaches attempt to extract local spatial patterns such as action units and their combinations. Real facial motion, however, is never completely localized. The designer of FACS, Ekman himself has pointed out some of these action units as an unnatural type of facial movements. Detecting a unique set of action units for a specific facial expression is not guaranteed. One promising approach for recognizing up to facial expression intensities is to consider the whole facial image as a single pattern. Kimura and his colleagues have reported a method to construct emotion space using 2D elastic net model and K-L expansion for real images [4]. Their model is user independent and gives some unsuccessful results for unknown persons. Later, Ohba proposed facial expression space employing principle component analysis, which is person dependent [5].

In this research, an expression vector relative to neutral facial image of a particular person has been constructed using Bézier curves approximation as features. To make a person specific modeling, apexes of primary facial expressions are used as references. The Facial Expression Space (FES) is constructed based on multidimensional scaling which will map an unknown input facial image

Manuscript received April 5, 2009
Manuscript revised April 20, 2009

relative to known reference images with the generalization capability.

2. Face detection algorithm

The face detection system contains four steps:

- The first step converts the image from the RGB space to YCbCr space.
- Classify each pixel in the given image as a skin pixel or a non-skin pixel using fuzzy logic.
- Then edge detection approach is used in order to generate the segmented image.
- In the last step, we use Bézier curves for the approximation of the features, especially two eyes and lips.

A. Skin Color Segmentation

Skin-color segmentation technique is considered as an effective tool for face detection because it is invariant to changes in size, orientation and occlusion [6]. In this paper, we propose to use the YCbCr color space for two reasons:

- By using YCbCr color space, we can eliminate as much as possible the variation of luminance component caused by the lighting condition.
- The YCbCr domain is extensively used in digital video coding applications.

YC_bC_r is a color space that separates the luminance from the color information. Luminance is encoded in the Y and the blueness in C_b and the redness in C_r. It is very easy too convert from RGB to YC_bC_r [6]:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ C_b &= -0.169R - 0.331G + 0.500B \\ C_r &= 0.500R - 0.419G - 0.081B \end{aligned} \quad (1)$$

B. Fuzzy Classification

In this step, each pixel of the image is classified into skin pixel and non-skin pixel. The most suitable arrangements that we found for all input images in database are: C_b in [77,127] and C_r in [139,210]. These arrangements are not sufficient to find a good classification because of the diversity of human skin color and for many other reasons such as luminance, noise and shad. To overcome these problems, we propose to apply a fuzzy approach for pixel classification, since fuzzy set theory can represent and manipulate uncertainty and ambiguity. In this work, we use the Takagi-Sugeno fuzzy inference system (FIS) [7]. This system is composed of two inputs (the two components C_b and Cr) and one output (the decision: skin or non-skin color); each input has three sub-sets (light, medium and dark). Our algorithm uses the basic concept of fuzzy logic called fuzzy IF-THEN rules; these rules are applied in each pixel in the image in order to decide whether the pixel represents a skin or non-skin region. The fuzzy logic rules applied for skin detection are as follows:

1. IF C_b is Light and Cr is Light THEN the pixel=0
2. IF C_b is Light and Cr is Medium THEN the pixel=0
3. IF C_b is Light and Cr is Dark THEN the pixel=0
4. IF C_b is Medium and Cr is Light THEN the pixel=0
5. IF C_b is Medium and Cr is Medium THEN the pixel=1
6. IF C_b is Medium and Cr is Dark THEN the pixel=1
7. IF C_b is Dark and Cr is Light THEN the pixel=0
8. IF C_b is Dark and Cr is Medium THEN the pixel=1
9. IF C_b is Dark and Cr is Dark THEN the pixel=0

The first step is to determine, for each input, the degree of membership to the appropriate fuzzy sets via membership functions. Once the input has been fuzzified, the final decision of the inference system is the average of the output (Zi) corresponding to the rule (r_i) weighted by the normalized degree of the rule (1 or 0). We note that the number of the skin pixels detected by fuzzy logic is more important than the number using classic classification. The detection of face region boundaries by such a segmentation process is illustrated in **Fig. 1.**

Fig. 1 Detection of face region by skin color segmentation.

3. Construction of Expression Change Model

The degree of facial expression change depends on the displacement of the FES from the neutral face [8]. So in this work, we have model the relationship between the

degree facial expression change and the displacement of the FES using the movement of the control points of a Bézier curve [9].

A. Bézier Curve

A Bézier curve $\mathbf{Q}(t)$ of degree n can be defined in terms of a set of control points $\mathbf{P}_i\,(i = 0,1,2,....,n)$ and is given by [6]:

$$\mathbf{Q}(t) = \sum_{i=0}^{n} \mathbf{P}_i B_{i,n}(t)\,, \qquad (2)$$

where each term in the sum is the product of a blending function $B_{i,n}(t)$ and a control point $\mathbf{P}_i$. The $B_{i,n}(t)$ are called Bernstein polynomials and are defined by

$$B_{i,n}(t) = C_i^n t^i (1-t)^{n-i}\,, \qquad (3)$$

where C_i^n is the binomial co-efficient given by:

$$C_i^n = \frac{n!}{i!(n-i)}. \qquad (4)$$

The order of the curve is fixed with constant 3. Some of the 3rd order Bézier curves are shown in **Fig. 2**. Movement of the control points influences the shape of the curve. The moving effect of control points is shown in **Fig. 3**. A Bézier curve is controlled by its control points $\mathbf{P}_i$ and a knot vector. In this work, Bézier curve has been used to describe the displacement of FES with the variable t denoting the degree of expression change. The control points are defined by the displacement of FES at specific t_i s, and the knot vector by the list of the degree of the facial expression change associated with each control point.

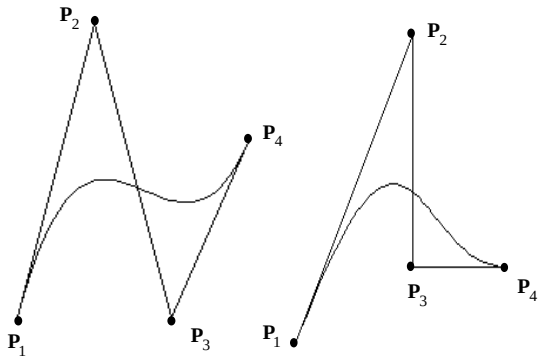

Fig. 2 3rd order Bézier curves.

B. Construction of Expression Change Models

For each category of facial expression, we have used 12 FES Bézier curves to construct the expression model. In this work, we have constructed the expression models for the six principal expressions: happiness, sadness, fear, anger, disgust and grief. In order to obtain the 12 average Bézier curves for an expression model, we have invited 5 experimental subjects.

For each subject, we have recorded a fast and slow image sequence with six expressions. In each sequence, the subject started from the neutral expression and changed smoothly until the apex of the expression. In total for each Bézier curve, we have 10 image sequences for averaging.

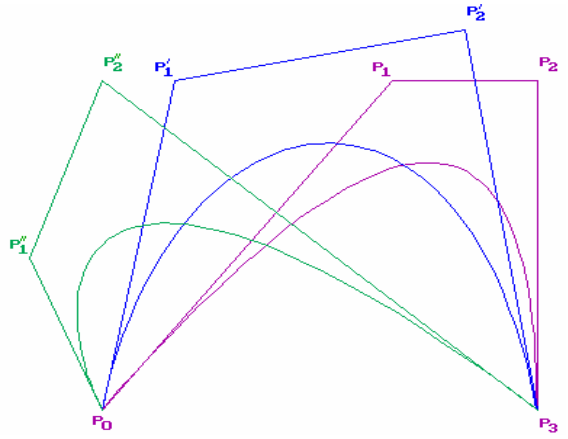

Fig. 3 Bézier curves with their control points.

C.1 Calculation of $\mathbf{Q}^s(t)$ from Image

$\mathbf{Q}^s(t)$ can be defined by the control points $\mathbf{P}_i^s$ and the knots t_i^s. Since the original displacement of FES $\mathbf{P}_i^s$ in each frame depends on each individual, it is necessary to normalize these displacements. First, we have normalized the dimension of images by the distance between the right and left eye inner corners whose locations are not subject to any expression change. A tilted face will be rectified to the upright position. Then we have used the width and height of eye brow, eye and mouth to normalize the displacement of FES. In order to determine the knots t_i^s from an image sequence with n frames, we have set the degree of expression change $t_1^s = 0$ at the start frame, and $t_n^s = 100$ at the apex frame. Since the degree of expression change t is directly proportional to the frame number of

the image sequence, the intermittent t_i^s in each frame can be determined.

C.2. Averaging of $\mathbf{Q}^s(t)$ from 10 Image Sequences

An average Bézier curve $\mathbf{Q}^*(t)$ from 10 image sequences of the same expression is obtained by calculating the new control points $\mathbf{P}_i^*$ at $t_i^* = (0,10,20,.....,100)$.

$$\mathbf{P}_i^* = \frac{1}{10}\sum_{s=1}^{10}\mathbf{Q}^s(t_i^*), i = (1,2,.....,11), \quad (5)$$

where $\mathbf{Q}^s$ are the Bézier curves. From $\mathbf{P}_i^*$ and t_i^*, we can calculate the $Q^*(t)$.

Now we construct the six expression change models, each of which consisting of 12 average Bezier curves $\mathbf{Q}^*(t)$ of FES. Each $\mathbf{Q}^*(t)$ describes the relationship between the degree expression change and the displacement of a FES. When we have an input of an expressional image sequence, we first track different FES, and then among them match the trajectory of FES with the six expression models to recognize the category and degree of expression change in the image sequence.

4. Tracking Facial Feature Points

The shade and appearance of facial features change when the observed subject changes his or her expressions. So tracking using only the ordinary template comparison is not as easy as it may be seen. In order to improve the reliability of the tracking system, FES tracking is viewed as a labeled graph matching problem as proposed by J. Buhmann, which fuses the FES template with knowledge about the geometric relationship between the FES. The initial location of the FES in the first frame is assumed to be known.

A. Labeled Graph

We treat FES as nodes in a labeled graph, with the labels being the template composed of 17x17 gray levels around the node. Neighboring FES are linked to form a topological graph, with each link consisting of the Alamin-Hama's apple-node distance between the FES. Furthermore, to preserve the local topology more rationally, we weight the links with a parameter related to facial characteristics. For example, as the mouth can deform violently, the template similarity for the mouth nodes should be regarded much more important than topological constraint in tracking. So we have given a smaller weight to the links between mouth nodes. In contrast, the nose does not deform very much in expression change. Thus we give larger weights to the links between the nose nodes. These weights are empirically determined.

The FES tracking system consists of two layers, a memory layer *M* and an input layer *I*. Each layer is constructed as a labeled graph. Correspondence FES between frame *n*-1 and frame *n* in image sequence is posed as a labeled graph matching problem, where frame *n*-1 is treated as the memory layer, and frame *n* as the input layer.

B. Cost Function

The graph matching is realized by minimizing a cost function. The cost function has two parts: a similarity measure between the sets of matched features, and a topology measure, which preserves the spatial relationship among matched features.

Let $\vec{T}_i^M = \{M_1, M_2, ..., M_{20x20}\}$ be the template of node i in layer M and $\vec{T}_i^I = \{I_1, I_2, ..., I_{20x20}\}$ the template of corresponding node in layer *I*, where $M_1, M_2, ..., M_{20x20}$ are 20x20 gray levels of FEPS *I* in frame *n*-1 and $I_1, I_2, ..., I_{20x20}$ in frame *n*. Resemblance between templates in the two layers is assumed by means of a similarity function

$S_n(\vec{T}_i^M, \vec{T}_i^I)$ defined by:

$$S_n(\vec{T}_i^M, \vec{T}_i^I) = \frac{\vec{T}_i^M \cdot \vec{T}_i^I}{\| \vec{T}_i^M \| . \| \vec{T}_i^I \|}. \quad (6)$$

Similarities are assumed over all pairs of corresponding nodes in the two layers, yielding a "similarity cost"

$$C_{simi} = -\sum_{i \in N} S_n(\vec{T}_i^M, \vec{T}_i^I), \quad (7)$$

where *N* is the set of the nodes in the labeled graph.

The topological constraint between two layers is defined by the sum of preservation quality S_l as follows:

$$C_{topo} = \sum_{(i,j) \in L} S_l(\vec{D}_{ij}^M, \vec{D}_{ij}^I)$$

$$= \sum_{(i,j)\in L} \alpha_{ij} (\vec{D}_{ij}^{M}, \vec{D}_{ij}^{I})^2 , \quad (8)$$

where $\vec{D}$ is the Euclidean distance between node *i* and node *j* in the same layer, α_{ij} the weight of the link between node *i* and node *j*, and *L* the set of links.
Obviously a good match will be one where both these costs are minimized concurrently. We merge them into one cost function as:

$$C_{total} = C_{simi} + \lambda C_{topo} . \quad (9)$$

The coefficient λ determined by experimentation controls the rigidity of the image graph.

C. Graph Matching

The graph matching is a dynamic process of node diffusion that minimizes the cost function based on a simulated annealing procedure. By using the position of FES and the expressional information obtained in frame *n*-1, we can choose a suitable initial placement of FES in frame *n*. Then we construct layer *I* and compute the initial cost. To minimize the cost, we can

1. choose a random node in layer *I* and shift it by a random displacement vector.
2. shift if either
   - the cost C_{total} is reduced due to this move, or
   - change in cost ΔC_{total} satisfies a probability $\exp(\Delta C_{total} / T)$, where T is the annealing temperature with a geometric cooling schedule.

This process of node diffusion will be repeated until the cost value does not change any longer. The obtained position of FES in frame *n* can be used to detect the degree of expression changes in this frame, and renew the layer *M*. The system will then be ready to process the next image *n*+1 in the sequence.

5. Recognition of expression

In this research work, the facial expressions have been recognized not only by a static image but also with video image sequences. By using the motion information, the facial expressions can be recognized more precisely and more reliably. When motion of FES has been detected in 3 consecutive frames, we could assume that the change of expression was started, and we could start to recognize the facial expression in the input image sequences. The frame before the first frame of the 3 frames was regarded as the start frame of expression change.

6. Experimental Results

The effectiveness and robustness of the algorithm is justified using different images with various kinds of expressions. Experiments are carried out on a Pentium IV 1.2GHz PC with 512 MB RAM. The algorithm has been implemented using visual C#.
We have implemented the 3rd order Bézier curves for the analysis of facial expressions. So line profiles from the image sequences have been extracted in this investigation and have been approximated with 3rd order Bézier curves, as shown in **Fig. 4**. The control points of these Bézier curves are being computed and are stored in the respective data files. In order to compare the facial expressions of image of a person with those stored in the database, different curves for the eyes and mouth are extracted and approximated with cubic Bézier curves. The control points of these approximated Bézier curves are then compared after affine transformation to compute the similarity between different expressions using Apple-node distance metric[7].

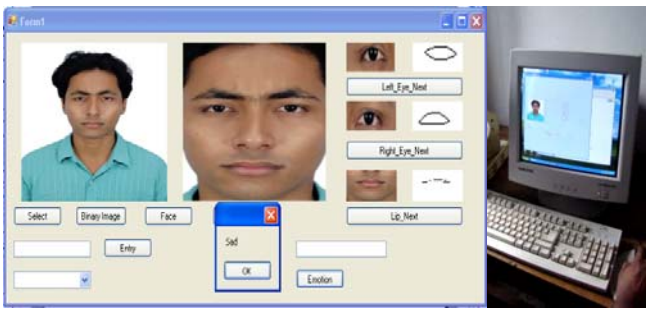

Fig. 4 (a) Facial expression recognition by extracting line profiles from face image and approximating with Bézier curves (b) Controlling the robot using the facial expression.

When a face image is subjected in the input, the facial expression analysis detection result highlights the facial part (movement of the controls of the Bézier curves) of the images. Finally these Bézier curves were analyzed for finding the change in facial expressions. Although six psychological expressions are distinct for representing human facial expression analysis, but we considered only normal expression, happiness, sadness, fear and surprise. In order to recover the control points of the Bézier curves, a number of 3rd order Bézier curves have been justified for this experiment and their control points have been found exactly, as shown in **Fig. 5**. The error versus learning time for the 3rd order Bézier curve is shown

Expression	Correct	Missed	Movements
Normal Face	50	1	Left-right
Happiness	44	6	Draw picture
Surprise	50	0	Bottom-up
Fear	47	3	Wrist up-down
Sadness	46	4	Shoulder move

graphically in **Fig. 6**, which reveals that the error is decreasing rapidly. Recognition results of the facial expression for 200 image sequences are given in **Table 1**. Experimental results reveal that success rate is better for surprise expression because the intermediate control points move appreciably when a user becomes surprised.

7. Implementation

A manipulator robot, connected to the pc through the parallel port, has been controlled by means of commands directed by the facial expressions of the user, as shown in **Fig. 4**. The robot has several movements, such as: Left-right, Draw picture, Forward, Backward, Top-down, Bottom-up and so on, depending on the facial expressions. Some of the expressions employed for controlling the robot are listed in **Table 1**.

8. Conclusion

This research proposes a new approach for recognizing the category of facial expression and estimating the degree of the continuous facial expression change from time sequential images. This approach is based on person-independent average facial expression models and precise FEPS tracking techniques. We have constructed the expression models by using average Bézier curves from several subjects. So we can recognize expression change of any subject. By preserving the local topology of FEPS and considering the facial characteristics, the proposed tracking system of labeled graph matching with weighted links we have tracked the FEPS precisely. By using the information of a total of 12 FEPS simultaneously to recognize facial expression, we can recognize the category of facial expressions robustly and estimate the subtle degree of expression change.

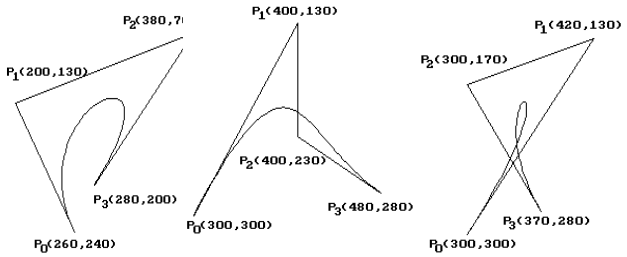

Fig. 5 Recovering the control points of the Bézier curves of the images.

Table 1 Recognition results for facial expression.

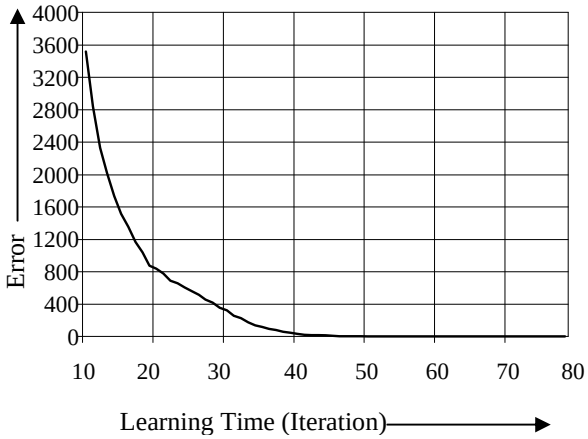

Fig. 6 Error versus learning time for Bézier curve.

In addition, we can minimize the node diffusion range to reduce the search time in tracking by feedback from the obtained expression information. In this investigation, four different facial expressions of 200 individual persons pictures have been analyzed. In this paper, 3rd order Bézier curve has been used to identify the face outlines and expressions. Simple calculation of the sum of the distances between the corresponding points takes huge amount of computational cost. The adoption of the cubic Bézier curves has reduced the computational cost, as only four control points are sufficient to represent a curve.

Although this method has been implemented for a few persons, but the experimental results nevertheless demonstrate that our system is reliable if the images represent a distinct view of the faces.

References

[1] Y. Yacoob and L.S. Davis, "Recognizing human facial expressions from long image sequences using optical flow", IEEE Trans. Pattern Analysis & Machine Intelligence, Vol. 18, No 6, pp. 636-642, 1996.

[2] P. Ekman and W. Friesen, "Facial Action Coding System", Consulting Psychologists Press, 1977.

[3] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications", Proc. IEEE, Vol. 83, No. 2, pp. 259-271, 1995.

[4] S. Kimura and M. Yachida, "Facial expression recognition and its degree estimation", Proc. Computer Vision and Pattern Recognition, pp. 295-300, 1997.

[5] K. Ohba, G. Clary, T. Tsukada, T. Kotoku, and K. Tanie, "Facial expression communication with FES", Proc. International Conference on Pattern Recognition, pp. 1376-1378, 1998.

[6] M.A. Bhuiyan and H. Hama, "Identification of Actors Drawn in Ukiyoe Pictures", Pattern Recognition, Vol. 35, No. 1, pp. 93-102, 2002.

[7] M. B. Hmid and Y.B. Jemaa, Fuzzy Classification, Image Segmentation and Shape Analysis for Human Face Detection. Proc. of ICSP, vol. 4, 2006.

[8] M. Wang, Y. Iwai, M. Yachida, "Expression Recognition from Time-Sequential Facial Images by use of Expression Change Model", Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 324 – 329, 1998.

[9] M. I. Khan and M. A. Bhuiyan, "Facial Expression recognition for Human-Machine Interface", ICCIT, 2006.