Face Recognition Using Interpolated Bezier Curve Based Representation

Sarbajit Pal[1], P.K.Biswas[2] and Ajith Abraham[3]
[1]*Variable Energy Cyclotron Centre, 1/AF Saltlake, Kolkata, India*
[2]*Dept. of E&ECE, Indian Institute of Technology, Kharagpur, India*
[3]*Department of Computer Science, Oklahoma State University, USA*
sarbajit@veccal.ernet.in, ajith.abraham@ieee.org

Abstract

This paper proposes an efficient face recognition scheme, which incorporates certain methods. In this scheme, edges are obtained from sobelled face images. The major feature areas are segmented and principal curves are obtained by applying thinning algorithm on the features of resulting face image. A modified thinning algorithm based on line sweep procedures is used. A line sweep is a process where the plane is divided into parallel slabs by lines passing through certain "events" and then items are processed according to an order of the slabs [1]. A new method for Bezier contour point interpolation is used after locating the two end points of the curve on the prime facial features. The two control points of a Bezier curve are interpolated by this method. The Bezier points i.e. the two end points of a curve and the two control points, form the line segments whose Hausdroff distances are calculated as specified in the algorithm discussed in this paper. The recognition rate is 100% on normal faces and also on faces with some simple expressions.

1. Introduction

All automated or computerized face recognition has attracted much interest over the past 20 years. Much work has been done during the past years to improve this technology. Such interest has been motivated by the growth in applications in many areas, including face identification in law enforcement and forensics, user authentication in building access or automatic transaction machines, indexing of, and searching for, faces in video databases, bankcard identification, access control, mug shots searching, intelligent user interfaces, etc. The application of face recognition technology can be categorized into two main parts: law enforcement application and commercial application. Face recognition technology is primarily used in law enforcement applications, especially mug shot albums (static matching) and video surveillance (real-time matching by video image sequences). The commercial applications range from static matching of photographs on credit cards, ATM cards, passports, driver's licenses, and photo ID to real-time matching with still images or video image sequences for access control.

The most popular approaches in the face recognition literature are mainly identified by their differences in the input representation. Two major input representations are used, namely the geometric feature-based approach and the image-based approach. The matching procedure of input and model faces used in the majority of the geometric or image-based approaches utilizes fairly standard distance metrics like the Euclidean distance and correlation.

The feature-based technique extracts and normalizes a vector of geometric descriptors of biometric facial components such as the eyebrow thickness, nose anchor points, chin shape and zygomatic breadth etc. The vector is then compared with, or matched against, the stored model face vectors.

2. Previous Work

The major human face recognition techniques that apply mostly to frontal faces are eigenface (eigen feature), neural etwork, dynamic link architecture, hidden Markov model, geometrical feature matching, and template matching. Eigenface is one of the most thoroughly investigated approaches to face recognition. It is also known as Karhunen-Loeve expansion, eigenpicture, eigenvector, and principal component. Sirovich and Kirby [2] and Kirby et al. [3] used principal component analysis to efficiently represent pictures of faces. They argued that any face images could be approximately reconstructed by a small collection of weights for each face and a standard face picture (eigenpicture). The weights describing each face are obtained by projecting the face image onto the eigenpicture. The authors reported 96 percent, 85 percent, and 64 percent correct

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
0-7695-2108-8/04 $ 20.00 © 2004 IEEE

classifications averaged over lighting, orientation, and size variations, respectively. Their database contained 2,500 images of 16 individuals. As the images include a large quantity of background area, the above results are influenced by background. In summary, eigenface appears as a fast, simple, and practical method. However, in general, it does not provide invariance over changes in scale and lighting conditions.

The use of neural network could be due to its non-linearity. Hence, the feature extraction step may be more efficient than the linear Karhunen-Loeve methods. The way in constructing a neural network structure is crucial for successful recognition. It is very much dependent on the intended application. Lawrence et al. [4] proposed a hybrid neural network, which combined local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network. The SOM provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimension reduction and invariance to minor changes in the image sample. The convolutional network extracts successively larger features in a hierarchical set of layers and provides partial invariance to translation, rotation, scale, and deformation. The authors reported 96.2 percent correct recognition on ORL database of 400 images of 40 individuals. The classification time is less than 0.5 second, but the training time is as long as 4 hours. There are also many other works done using neural networks, which is beyond the discussion of this paper.

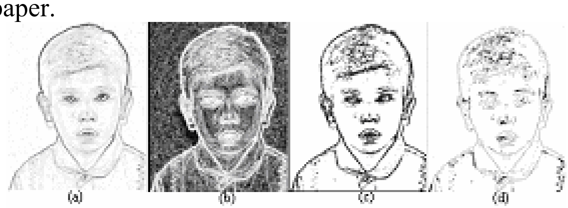

Figure 1. (a) edge (b) histogram equalized (c) binarised and (d) thinned image

3. The Proposed Method

However, the purpose of our work has been recognition of faces using a slightly different technique than the techniques discussed before. Gray-scale or colored images are taken as input to the system and edges are obtained from preprocessed image as shown in Fig 1. The face images of many people in normal and with simple expression (smiling and angry) condition are considered. The facial area is separated from the head and shoulder scene in order to fit the parameterised face model to the speaker. These images are masked, so as to remove the background features.

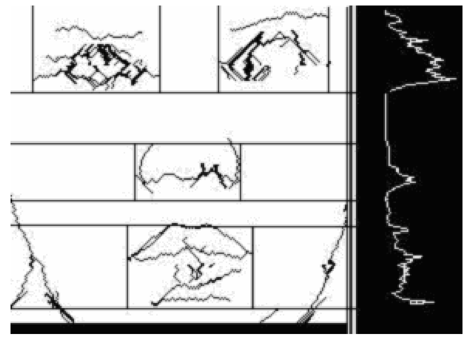

Figure 2. Detection of major features regions by vertical followed by horizontal projection

The mask is chosen in a way so that it extracts only the portion of the face consisting of eyebrows, eyes with eyeballs, nose boundary, lip-boundaries and lower jaw boundary (Fig 2). The masked face images are transformed using the sobel operator to obtain the edges. Once, the edges are determined, the modified thinning algorithm is applied to thin the edges obtained. Principal lines are obtained by applying thinning algorithm over the required features. The contours of eye, nose and mouth are extracted by deformable template matching [5][6][7]. To represent these curves as Bezier curves, two control points for the Bezier curve are required. As already known, a Bezier curve is made of four points where two points are located at the two end points of the curve. The other two control points are obtained by Bezier control point's interpolation method as discussed later. These Bezier points for each curve form three adjacent straight-line segments. The co-ordinates of the Bezier points are obtained through a program. Finally, these line segments are used in calculation of Hausdroff distance.

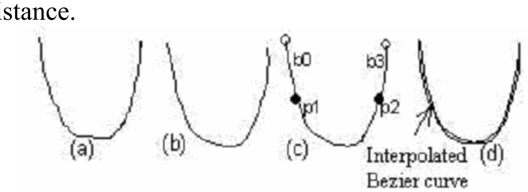

Figure 3. (a) face contour (b) after smoothing operation (c) four chosen points for t=0, ¼, ¾, 1 (d) given contour and interpolated Bezier curve

The Hausdroff distance [8] measured for line segments consisting of spline points provide a better recognition rate than the one measured for line segments of a LEM [9]. A model database consisting of spline point line segments is created for all face images in normal condition. A test database consisting of spline point line segments of a test image with expression is created for comparison with the model database by calculating Hausdroff distance. The set of line segments representing a face in the model for which the minimum Hausdroff distance is identified while comparing with the test set of line segments, is considered to be the most similar to the test face.

4. Modified Thinning Algorithm

A modified thinning algorithm based on line sweep procedures is used. A line sweep is a process where the plane is divided into parallel slabs by lines passing through certain "events" and then items are processed. This algorithm thins the edge image keeping the line segments, which travel across the boundary and removing the lines of smaller length and isolated regions. The major facial features (eyes, nose and lips) are located from the horizontal and vertical projection of the thinned image as shown in Fig.2.

(Step 1) Binarise by determining the threshold of histogram equalised edge image and set the initial window size to NxN pixels.

(Step 2) Using the current window size, scan the whole image starting from the top left corner and slide the window at N pixels horizontally and vertically until the whole image has been covered.

(Step 3) At each location, scan the top, right, bottom and left sides of the window and note the middle of the edge points (P_i) present on the whole window boundary. For each Pi follow the edge towards its opposite boundary.

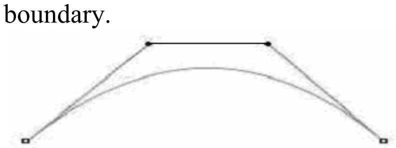

Figure 4. Beizer curve represented with 3 line segments

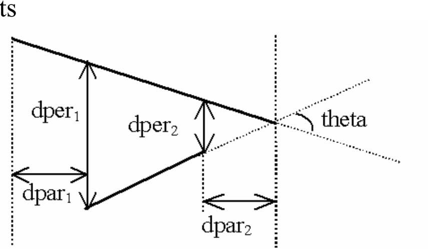

Figure 5. Line displacement measures

(Step 4) For thinning to be performed, find out the width of the edge and keep a pointer at the middle point. Mark the entire width by '0' and middle point by '1'.

(Step 4a) Shift the pointer to the nearest point on the next row or column as the edge is followed horizontally or vertically respectively. Count the number of '1's on the either side of the pointer including the location of the pointer, extended upto the edge width, which is dynamically varied. In case the of unequal count shift the pointer by one pixel towards the greater count and repeat step 4a till a valid count is obtained at shifted pointer.

(Step 5) Repeat the step 4 until all P_i's are covered.

5. New Method for Contour Interpolation

The given interpolation method finds out other two Bezier control points of the given curve where starting and terminating points are known. It is required to know four points on a given curve with respective t to interpolate two intermediate control points. A curve can be completely generated from its parametric form by varying t from 0 to 1. Therefore as t varies from 0 to 1/4 and 0 to 3/4, 25% pixel and 75% pixel of the curve is generated respectively. The bending functions representing the Bezier polynomial have maximum bend at t=1/4 and t=3/4. Here we consider b0 and b3 are known end points of a curve whose other two Bezier control point is to be estimated. To compute the other two control points, we find out two more points located at the end of 25% and 75% of the full segment as shown in fig.5. Thus p1 and p2 can be written in terms of four Bezier control points as follows.

$p1 = 1/64(27b0 + 27b1 + 9b2 + b3)$

$p2 = 1/64(b0 + 9b1 + 27b2 +27 b3)$

The values of b1 and b2 can be approximated by solving the parametric equations formed by taking t=1/4 and t=3/4 respectively.

$b1 = 8/9(3p1 - p2) – 1/9 (10 b0 - 3b3)$

$b2 = 8/9(3p2 - p1) + 1/9 (3 b0 - 10b3)$

In case of feature boundary extraction, each segment is chosen such a way that each segment must contain only one bend.

6. Hausdroff Distance of Line Segment

Hausdroff Distance is designed to measure the similarity between line segments generated from Bezier curve points. The Bezier curves are drawn over principal lines of faces. Each principal curve is determined as a set of three adjacent line segments i.e. consisting of four points as in Fig.4. The principal lines considered here, constitute a set of fifteen line segments, i.e. a total of forty-five line segments. A line segment database for all normal faces is created. This forms the model database. Each line segment consists of four integer values which denote the end point co-ordinates of each line segment i.e x_1, y_1, x_2, y_2 respectively for the end-points (x_1,y_1) and (x_2,y_2). Consider a set $M^s = \{m^s_1, m^s_2, m^s_3, m^s_4, \ldots\ldots, m^s_k\}$ to be the model consisting of line segments m^s_1, m^s_2, m^s_3. A test set $T^s = \{t^s_1, t^s_2, t^s_3, t^s_4, \ldots\ldots, t^s_k\}$ consists of the line segments of the test face image to be compared with the set of model normal face images. The Hausdroff distance between line segments is actually based on a vector *dls* that represents the distance measure between any pair of line segments, one each from the model and the test set. The vector is defined as $\mathrm{dls} = [\mathrm{dper}, \mathrm{dpar}, \mathrm{dori}]$, where *dper*, *dpar* and *dori* are the perpendicular distance, parallel distance and the orientation distance respectively as shown in Fig.5, between the same pair of line segments from the model and test sets. All these entries are defined as d*ori* = *f*(*theta*), *dpar* = *min*($pard_1$, $pard_2$) and *dper* = perpendicular distance between two line segments. '*theta'* computes the intersecting angle between the two line segments. *f*(*theta*) is a function that converts the angle value into a scalar quantity. Here, we have taken *f*(*theta*) = $theta^2$/W. W is a weight that is carefully chosen by training process to be the length of the test line segment. However, the value of *theta* is calculated using some basic mathematics of 2D co-ordinate geometry. First, the slopes of the model and test line segments are calculated as $slm = (ym_2 - ym_1) / (xm_2 - xm_1)$ and $slt = (yt_2 - yt_1) / (xt_2 - xt_1)$ respectively. Where, (xm_1, ym_1), (xm_2, ym_2) and (xt_1, yt_1), (xt_2, yt_2) are the two end point co-ordinates of the model and test line segments respectively. The Fig.4 Beizer curve represented with 3 line segments n *theta* is calculated as $\tan^{-1} \frac{(\mathrm{slt} - \mathrm{slm})}{1 + (\mathrm{slt} \times \mathrm{slm})}$. With the value of theta, $dori = (theta^2) / lent$, where '*lent'* is the length of the test line segment. If the value of *theta* is neither 0° nor 180°, we assume that the line segments are not parallel to each other. In such a case, *dper* is taken as the minimum of the perpendicular distances $perd_1$ and $perd_2$. These perpendicular distances are computed as

$$\mathrm{perd}_1 = \sqrt{(\mathrm{yt}_1 - \mathrm{ym}_1)^2 + (\mathrm{xt}_1 - \mathrm{xm}_1)^2}$$

$$\mathrm{perd}_2 = \sqrt{(\mathrm{yt}_2 - \mathrm{ym}_2)^2 + (\mathrm{xt}_2 - \mathrm{xm}_2)^2}$$

If the lines are parallel to each other, the perpendicular distance is the distance of the perpendicular dropped from any point of the test line segment on the model line segment. If we consider the equation of the model line segment to be $y = slm \times x + yintm$ in two unknowns, *yintm* is calculated as $((xm_2 \times ym_1) - (xm_1 \times ym_2)) / (xm_2 - xm_1)$. If the perpendicular is dropped from point (xt_1, yt_1) of the test line segment, the distance of the perpendicular is given by

$$\mathrm{dper} = \frac{|\mathrm{yt}_1 \ \ (\mathrm{slm} \times \mathrm{xt}_1) \ \ \mathrm{yintm}|}{\sqrt{1 + \mathrm{slm}^2}}$$

The parallel distance *dpar* is defined as the minimum displacement to align either end points of the test line segment with the model line segment. Two perpendiculars are dropped from either end points of the test line segment on the model line segment. The co-ordinates for the feet of the perpendiculars are denoted by (xmt_1, ymt_1) and (xmt_2, ymt_2) respectively, as perpendiculars from points (xt_1, yt_1) and (xt_2, yt_2). The feet of the perpendiculars are calculated as

$xmt_1 = ((slm \times yt_1) + xt_1 - (slm \times yintm)) / (slm^2 + 1)$.
$ymt_1 = ((slm^2 \times yt_1) + (slm \times xt_1) + yintm) / (slm^2 + 1)$.
$xmt_2 = ((slm \times yt_2) + xt_2 - (slm \times yintm)) / (slm^2 + 1)$.
$ymt_2 = ((slm^2 \times yt_2) + (slm \times xt_2) + yintm) / (slm^2 + 1)$.

The parallel distances $pard_1$ and $pard_2$ are calculated using the distance formula

$$\mathrm{pard}_1 = \sqrt{(\mathrm{ymt}_1 - \mathrm{ym}_1)^2 + (\mathrm{xmt}_1 - \mathrm{xm}_1)^2}$$

$$\mathrm{pard}_2 = \sqrt{(\mathrm{ymt}_2 - \mathrm{ym}_2)^2 + (\mathrm{xmt}_2 - \mathrm{xm}_2)^2}$$

We consider the minimum of $pard_1$ and $pard_2$ as *dpar.* Finally, the distance between the two line segments is defined as $\mathrm{dls} = \sqrt{\mathrm{dori}^2 + \mathrm{dpar}^2 + \mathrm{dper}^2}$

The primary line segment Hausdroff distance is defined as *pLHD* = *max* (*hMT, hTM*), where $\mathrm{hMT} = \frac{1}{\sum \mathrm{lenm}} \sum \mathrm{lenm}.\min \mathrm{dm}$, *lenm* is the length of model line segment, *mindm* is the minimum *dls* among a set of test line segments and equal number of model line segments. $\mathrm{hTM} = \frac{1}{\sum \mathrm{lent}} \sum \mathrm{lenm}.\min \mathrm{dm}$, *lent* is the length of the test line segment. The *HpLHD* is computed between the set of test line segments and each such set of model line segments. The minimum of the *HpLHD* is calculated and defined as *minhaus* (minimum Hausdroff distance). The model set of line segments with the minimum Hausdroff distance is considered as the face image, which is most similar with the test face image.

7. Experiment Results

In order to evaluate the performance of the Hausdroff Distance algorithm, we used the Yale Face database [10]. The Yale face database consists gray scale images of fifteen subjects. The model face database is created using these 15 subjects' images in normal, smiling and angry condition. Fig 6 denotes the subject from Yale face database in normal condition and happy expression.

Testing faces	Eigenface		EM	LEM	LBP
Smiling Expression	20-eigenvectors	87.85%	52.68%	78.57%	100%
	60-eigenvectors	94.64%			
	112-eigenvectors	93.97%			
Angry Expression	20-eigenvectors	78.57%	81.25%	92.86%	100%
	60-eigenvectors	84.82%			
	112-eigenvectors	87.50%			

Table 1. Comparison of recognition results

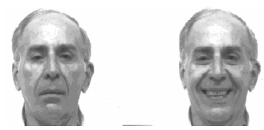

Figure 6. Yale face data (normal and happy)

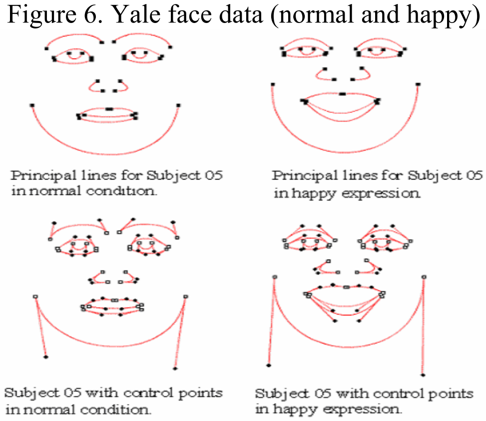

Figure 7. Bezier curve and control point interpolation

The expressions such as happy, sad, surprised and faces with spectacles are considered for the experiment of face recognition. The faces with expressions are compared against the model face database consisting of normal faces. All the face images are normalized using some parameters discussed before. The Bazier points are interpolated over the principal lines of a facial features. These points for each curve form three adjacent line segments. The Hausdroff distance is calculated based on these line segments. The efficiency of the algorithm is 100%, which indeed conforms to the statement that Hausdroff distance is a nobel measure in measuring similarity in faces. Table 1 shows the comparison of recognition results under different facial expressions using methods of Eigenface, EM, LEM and LBP (Line through Bezier points).

8. Conclusion

In this paper, a modified thinning algorithm, a distance measurement algorithm has been presented. The advantage of thinning is that the principal curves can be obtained distinctively from the face image. Thinning of edges obtained from the sobelled image can eliminate unnecessary lines and noise. The advantage of using Bezier curve based model is that it has relatively few control points. Also, the Hausdroff distance similarity measure takes the parallel, perpendicular and orientation distances into consideration. The system is not designed for the faces having beard and moustache. However, the system can distinguish the same subject with different facial expressions as happy, surprised, and sad.

References

[1] F. Chang, Y. Chang, T. Pavlidis and T. Shuai, "A Line Sweep thinning Algorithm", *IEEE,* 1995.

[2] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterisation of human Faces," *J. Optical Soc. Of Am.,* vol. 4, pp. 519-524, 1987.

[3] M. Kirby and L. Sirovich, "Application of the Karhunen-Loe¡ve Procedure for the Characterisation of Human Faces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, pp. 831-835, Dec. 1990.

[4] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural-Network Approach," *IEEE Trans. Neural Networks,* vol. 8, pp. 98-113, 1997.

[5] A. L. Yuille, D. S. Cohen, and P. W. Hallinan. "Feature Extraction from Faces Using Deformable Templates, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'89),* IEEE Computer Society Press, pp. 104-109, 1989.

[6] Chung-Lin Huang and Ching-Wen Chen, "Human Facial Feature Extraction for Face Interpretation and Recognition," *Pattern recognition,* vol. 25, no. 12, 1992.

[7] S. Pal and P. K. Biswas, "Real time Video Tracking by Deterministic Template Matching," *International Conference on Communications, Computers and Devices (ICCCD-2000),* 2000.

[8] S. Pal and P. K. Biswas, "Modifiied Hausdroff Distance Transform Technique for Video Tracking," *International Conference on Vision, Graphics and Image Processing (ICVGIP-2000),* 2000.

[9] Y. Gao and Maylor K.H. Leung, "Face Recognition Using Line Edge Map", IEEE Transaction on pattern anlysis and Machine Intelligence, vol. 24, no. 6, 2002

[10] Yale Univ. Face Database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html, 2002.