Eye Detection and Face Recognition using Line Edge Maps

Mihir Jain¹, Suman Mitra², Naresh Jotwani³
¹ IIIT, Hyderabad, 500032
² & ³ DA-IICT, Gandhinagar, 382007

Abstract - The task of face recognition has been actively researched in recent years, and many techniques have been proposed. This paper reports on the design, implementation and experimental study of an improved technique for face recognition, including eye detection in the preprocessing phase of a localized face image. After a brief summary of some known face recognition techniques, we present an innovative algorithm for eye detection on gray intensity face image, which combines a feature-based approach using line edge map (LEM), geometrical properties of eyes, and the use of LEM eye templates. Matching in each case is carried out using primary line segment Hausdorff distance. Experimental results of face recognition using this technique are presented. A potential enhancement is also proposed, namely facial feature block-wise implementation of the LEM technique.

Index Terms – Line edge map, Eigenface, Bayesian face recognition, Eye detection, Line segment Hausdorff distance.

1. INTRODUCTION & MOTIVATION

Face recognition – i.e. automatically identifying a person from digital image -- is an important research problem spanning numerous fields and disciplines. Recognition is carried out by comparing selected facial features against a facial database. Over many other biometric systems, it has the benefits of being a passive and non-intrusive system [1].

Like any Pattern Recognition problem, Face Recognition is also a three-step process: (i) Segmentation (face detection), (ii) Feature Extraction and (iii) Classification. Generally, segmentation is referred to as face detection, and the latter two as face recognition. This paper deals with the latter two, along with the required preprocessing, which includes scaling and orientation correction of localized face image and location detection of facial features.

Much research has been carried out in this field and many different techniques have been proposed. Some of them are insensitive to some kinds of variation and others to some other kinds; for example, the Eigenface approach performs well with varying facial expressions, but badly with varying light [1]. Similarly, other existing techniques are also not free from some or other limitations. Further efforts are required to improve the performance of face recognition techniques, especially in the wide range of environments encountered in the real world.

The aim of the present work is to take a step forward towards improving the performance of some of the existing techniques. After studying some of the well-known techniques for face recognition, we
implement the Line Edge Map (LEM) technique for face recognition, with the required preprocessing, and test the technique on a standard face database.

The following technical challenges are to be tackled by any face recognition technique:

- **Varying lighting conditions** – Due to strong reflection on the face skin caused by illumination, the shape information on faces are suppressed or lost, which could result in the increase of the error rate of classification [1].

- **Varying facial expressions** – Varying expressions produce physical variations from the neutral expression, and consequently most facial features get distorted.

- **Varying pose** – When the system is tested on images with different poses, such as like looking up, looking down, _et cetera_, the recognition rate tends to decrease [1].

A suitable face feature representation, Line Edge Map (LEM), was proposed by Gao and Leung [10], which extracts as features line segments from a face edge map. LEM integrates the structural information with spatial information of a face image by grouping pixels of face edge map to line segments. After thinning the face edge map, a polygonal line fitting process known as the dynamic two-strip algorithm [11] is applied to generate the LEM of a face.

Gao and Leung also introduced the use of Hausdorff distance [14] to measure the similarity of face LEMs, basically using it as a shape comparison measure by defining the ‘distance’ between two line sets.

2. **BRIEF SURVEY OF SOME OTHER TECHNIQUES**

2.1 **Face Recognition Techniques**

_1. Eigenface Approach_

In mathematical terms, eigenfaces are the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images [2,3]. The eigenvectors are ordered to represent different amounts of variation among the faces. Each face is represented by a linear combination of the eigenfaces; for reducing time complexity, the face can also be approximated using only the ‘best’ eigenvectors with the largest eigenvalues. The best M eigenfaces define an M-dimensional space, i.e., the ‘face space’. Using a database containing 2,500 images of 16 individuals, success rates of this technique have been reported as 96 percent, 85 percent, and 64 percent correct classifications averaged over lighting, orientation, and size variations, respectively.

_II. Line Edge Map (LEM)_

Takács in [4] argued that edge images can be used for the recognition of faces without the involvement of high-level cognitive functions. A Line Edge Map approach, proposed in [1], extracts lines from a face edge map as features. Line Edge Map integrates the structural information with spatial information of a face image by grouping pixels of face edge map to line segments. After thinning the edge map, a polygonal line fitting process known as ‘dynamic two-strip algorithm’ [5] is applied to generate the LEM of a face. An example of a human frontal face LEM which we obtained by implementing this algorithm is shown in Figure 1. The basic unit of LEM is the line segment grouped from pixels of an edge map. The classification is done by similarity measurement using line segment Hausdorff distance [14].
III. Bayesian Approach

In Bayesian approach [6], a probabilistic similarity measure is used, which is based on Bayesian belief that the image intensity differences between two face images $I_1$ and $I_2$, denoted by $\Delta = I_1 - I_2$, are characteristic of typical variations in appearance of an individual. In particular two classes of facial image variations are defined -- intra-personal variations $\Omega_I$ (corresponding to different facial expressions of the same individual) and extra-personal variations $\Omega_E$ (corresponding to variations between different individuals). The similarity measure between two face images is then expressed in terms of a conditional probability, and Bayesian inference based on maximum a posteriori probability (MAP) or maximum likelihood (ML) is employed for face recognition.

IV. Using Edginess Map of Image:

To extract the edginess map of an image, a 1-D Gaussian filter $g(x)$ and its first order derivative $c(y)$ is proposed in [7]. The horizontal components of edginess are derived by first applying 1-D Gaussian filter $g(x)$ along each row of image:

$$h(x,y_r) = k(x,y_r) \ast g(x)$$

and then by applying $c(y)$, the first order derivative of $g(x)$, along each column of $h(x,y)$:

$$f(x_c,y) = h(x_c,y) \ast c(y)$$

Similarly the vertical component is obtained. Both the components are added to get the edginess map of the original image.

2.2 Preprocessing Techniques

I. Face Detection, Localization and Orientation Correction

Given an arbitrary image, the goal of face detection is to determine whether there are any faces in the image and, if present, to return the location and extent of each face [8]. Face localization is a simplified detection problem with the assumption that an input image contains only one face. Orientation correction is carried out on a localized face image by rotating it such that the line joining two eyes becomes horizontal. Before face recognition, it is necessary to perform (a) orientation correction and (b) scaling to bring all images to the same resolution.

II. Eye Detection

Technically it is advantageous to perform eye detection before the detection of other facial features, for the simple but important reason that the position of other facial features can be estimated using eye
position [9]. In addition, the size, location and image-plane rotation of the face in the image can be normalized by using the positions of the two eyes.

Eye detection methods can be broadly classified into three categories: feature based methods, appearance based methods and template based methods [9].

Feature based methods explore certain eye characteristics -- such as edge and intensity of iris, the color distributions of the sclera and the flesh -- to identify some distinctive features around the eyes. Although these methods are usually efficient, they lack accuracy for the images that do not have high contrast.

In template based methods, a generic shape-based eye model is designed. This template is then matched to the face image pixel by pixel to find the eyes. Such methods can detect the eyes accurately, but they are usually time-consuming. In order to improve the efficiency of this method, a technique has been proposed in [9] that first roughly detects the two regions of eyes using a feature based method, and then performs template-matching on the reduced area, giving in effect a hybrid technique.

Appearance based methods detect eyes based on their photometric appearance. These methods usually need to collect a large amount of training data, representing eyes of different individuals under different conditions. This data is used to train some classifier, and detection is achieved by classification.

After detecting the eyes, other facial features can be located using the Golden Ratio (φ), since many key physiological proportions of the human face are based on this ratio [10].

The rest of this paper is organized as follows:

Section 3 describes the proposed LEM & template-based algorithm for eye detection, with experimental results of eye detection as well as the subsequent face detection. A ‘block-wise’ implementation of the LEM technique is presented in Section 4, with results, while Section 5 summarizes the present work.

The test set for all the experiments carried out consists of 112 localized face images from the CalTech face database [11], which were randomly selected out of 450 images in the database.

3. LEM & TEMPLATE-BASED ALGORITHM

In the presence of many promising face detection methods [1], we assume that the face region in the input image has been roughly localized, given that the roughly localized face image may also contain some background. An original image from [12], shown in Figure 2(A), was manually cropped to get the roughly localized face image shown in Figure 2(B).

(A) Face image from [12] (B) Roughly localized face image

Figure 2
Our proposed algorithm is tolerant to some amount of background, and therefore the preceding process of face detection need not be ‘perfect’. If necessary, it may allow an increase in the detected face extent to ensure that complete face is included.

The following algorithm outlines the proposed LEM & template-based eye detection method:

**Algorithm for eye detection:**

Input: Roughly localized face image.
Output: Two eyes, eye centers, iris centers

// Feature based part
**Select potential regions for eyes.**
Make pairs of selected regions.
For each pair:
  Begin
    Perform orientation correction according to the pair.
    Test the pair 1) geometrically, 2) for distinctive features and 3) for symmetry.
    If all the tests are passed
      Store the pair as a potential region pair (PRP).
      If a sufficient number of PRPs are obtained
        Break.
  End

// Template matching
For each PRP:
  Begin
    Generate eye LEM template and perform matching
  End
Select the best matched PRP as the eye pair.
Use recursive centroid finding to get eye centers.
Find iris centers using eye centers.

Details of the various highlighted steps of this algorithm can be found in [13].

To get the part of face from eyebrow to chin and between two ears we use estimated eye centroids and \( \phi \). Because the two iris may not be in the centers of eyes, we extract the two eye regions and find the difference between horizontal coordinates of their centroids. This difference is now used as intra-ocular distance and the final face part is cropped.

The template used here is the LEM of an artificial pair of eyes including eyebrows, constructed based on \( \phi \) [10], as shown in Fig. 7. The matching is done by line segment Hausdorff distance.

![Artificial eye LEM](image)

**Figure 3**
Experimental Results

With the test set and the method of computation of normalized error [12], the eye detection algorithm gave 100% result for normalized error less than 0.1, and 94.6% result for normalized error less than 0.07.

Subsequent face recognition gave 97.5% result with this database.

4. BLOCK-WISE & I-LEM IMPLEMENTATION

When all other features except eyes, eyebrows, nose and lips (with some part below) are removed from the face LEM, what remains is an I-shaped face LEM which is here referred to as I-LEM of face. Experiments were conducted in which I-LEMs were used instead of face LEM, the rest of the algorithm being essentially unchanged.

In block-wise implementation, the face LEM (or even I-LEM) is divided into three blocks, containing respectively the eye region, nose region and lip region. Now the LEMs of the corresponding parts of each test face are compared with those of each model face. Eye region is extracted using $\phi$, but the nose region and lip region cannot be extracted by using $\phi$ alone, since the face may have a slight upward or downward tilt. Horizontal projection is performed on all the pixels in the region containing nose and lips to find the line between nose and eyes; here the global minimum should lie somewhere between nose and lips.

Extracted I-LEM of a face and its blocks are shown in Figure 4.

Experimental Results

Experiments were performed with (i) I-LEM, (ii) block-wise LEM and (iii) block-wise I-LEM. But the results obtained were essentially the same as those of the previous algorithm.

So, to compare these different algorithms, the experimental setup was made more stringent. Now, the 40 test images in first experiment are used as model images and 72 model images as test images. This makes recognition more difficult because:

(a) We have a smaller number of model images of each individual to find a match, and
(b) The 40 earlier test images (which are now model images) are having larger variations.

As an example, Figure 5 shows the model images of one subject.

For block-wise algorithms, the decision was made by polling. Weights given to face LEM (or I-LEM), eyes, nose and lip regions were in the ratio 2:2:1:1 respectively.
The results are presented in Table I.

<table>
<thead>
<tr>
<th>Method</th>
<th>Correctly detected faces</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face LEM</td>
<td></td>
<td>59</td>
</tr>
<tr>
<td>I-face LEM</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>Face LEM block-wise</td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>I-face LEM block-wise</td>
<td></td>
<td>64</td>
</tr>
</tbody>
</table>

Table I
Face recognition
Results of different implementations of LEM technique

With the same preprocessing and eye detection steps, the result for the earlier face LEM decreased from 97.5% to 81.94% in this new setup, which is intentionally made tougher to compare different methods.

Interestingly, the I-face LEM method gives better result than face LEM, perhaps because it is not affected by the reflection on the cheeks which can be misleading in variable illumination. Implementing face LEM block-wise also improves the performance from 81.94% to 87.5%.

Three model images of a subject - from [11]
Figure 5

5. CONCLUSIONS
After a study of some well-known face recognition techniques, an LEM-based technique is chosen for implementation. The technique is implemented successfully, as seen in the experimental results. In its preprocessing phase, the technique makes use of hybrid LEM feature-based and template-based eye detection. The eye detection algorithm produces 100% result on the CalTech database [11], for a normalized error of 0.1.
The face recognition technique is also implemented block-wise with face LEM and I-face LEM. Face recognition results of I-LEM and block-wise implementation show an improvement of 6-7 percent, and thus look promising.

REFERENCES


