

AN IMPROVED APPROACH TO THE LINE-BASED FACE RECOGNITION

Anlong Ming, Huadong Ma

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia
Beijing University of Posts and Telecommunications, Beijing 100876, China
anthony ming@gmail.com

ABSTRACT

The line-based face recognition method is distinguished by its features, but its development and application is limited to some inherent drawbacks. This paper propose a method for decreasing the influence under variable illumination intensity by using the line-based singular value (LSV) feature vector instead of image gray-level value to calculate “distance” between two lines. We prove that our method is invariant to the illumination intensity. Finally, we suggest a distributed computing algorithm using grid computing to solve the multi-scale computation. Experimental results show our approach is effective.

1. INTRODUCTION

Many methods of face recognition have been proposed during the past 30 years, such as PCA, LDA, HMM [5] and elastic matching [6]. PCA and LDA suffer the re-computation problem when a new face view is added. HMM spends lots of expense in training. Elastic matching requires great computational effort. Furthermore, all these methods are not very suited to parallel and distributed computing because a recognition task is difficult to be divided into subtasks when they work.

O.de Vel and S.Aeberhard proposed a novel image-based face recognition algorithm [1, 2, 3, 4] that uses a set of random rectilinear line segments of 2D face views as the underlying image representation, together with the nearest-neighbor classifier as the line matching scheme. The combination of 1D line segments exploits the inherent coherence in one or more 2D face views in the viewing sphere.

The above algorithm is well-suited to parallel and distributed computing because a face recognition task can be divided into many subtasks of calculating the “distance” between lines. Moreover, it is particularly effective in obtaining a high recognition rate performance (nearly 100% recognition rate on Oracle Research Laboratory (ORL) face database

[1]). The algorithm has some other distinguished advantages: Due to the randomized sampling of the image, the algorithm is robust to rotations of the face in the plane, changes in facial expressions and the presence or absence of glasses or other accessories. Multiple views are even better suited to 1D line-based algorithm. Also, the algorithm is better able to handle head rotations out of the plane than 2D view-based algorithms. Since the lines run from one face-boundary to another and have fixed dimensionality, the algorithm is also scale-invariant.

Unfortunately, the line-based algorithm has inherent drawbacks which make against its development and application since it has been proposed, and so far, these problems remain unsolved as follows:

- It is sensitive to changes in illumination intensity because the gray-level value is directly used.
- The computation is very huge. Olivier de Vel and Stefan Aeberhard select $N = 400$ in training, which means drawing 400 lines in a face image. It is far more complex than that face image itself and the expense of computation in drawing, scaling and recording a line of any angle is very high.
- The algorithm depends on the assumption that the face detection has been undertaken prior to the execution of the line-based algorithm and the face boundaries are available.

So, in this paper, for changes in illumination intensity, the LSV feature is used instead of gray-level value. For training and recognition, a distributed algorithm based on grid computing is used.

The remainder of the paper is organized as follows: the line-based face recognition algorithm is described in Section 2 and our approach is presented in Section 3. Experimental results and analysis will be given in Section 4. We conclude our work in Section 5.

2. RELATED WORK

In this section, we briefly outline the line-based face recognition algorithm. Given K face class $F_k (k = 1, 2, \dots, K)$, the aim is to recognize one of the K faces from one or more test face views.

The work is supported by the National Natural Science Foundation of China (90612013), the Co-sponsored Project of Beijing Committee of Education (SYS100130422), the Specialized Research Fund for the Doctoral Program of Higher Education (20050013010) and the NCET Program of MOE, China.

An image P is modeled as a regular lattice of $w \times h$ pixels. We first classify the pixels in P into two classes, $C_{p,1}$ and $C_{p,2}$. Class $C_{p,1}$ consists of the background pixels in the image P , and class $C_{p,2}$ consists of all those pixels that represent one or more faces in P . We are interested in the pixels in $C_{p,2}$ with neighbors in $C_{p,1}$ and call the set of the face boundary pixels β .

Consider l pixel values extracted along a straight line between two points in the image. For any two points $B_1 \in \beta$, $B_2 \in \beta$ in a face view V_k , such that the Euclidean distance between B_1 and B_2 is greater than the minimum D_{min} , let

$$L(B_1, B_2) \equiv (L^{(1)}, L^{(2)}, \dots, L^{(l)})$$

be a vector of length l , where l is the number of equi-spaced connected intensity values denoted by $L^{(q)}$ ($q = 1, 2, \dots, l$) along the image rectilinear segment from B_1 to B_2 . The line segment length l is a constant parameter determined *a priori*; larger values of l result in better classification rates at the expense of increased processing times. All lines are scaled to the value l by pixel interpolation. We call $L(B_1, B_2)$ a lattice line, denoted by L . Figure 1 illustrates how to obtain a line L with the length $l = 6$. Each shaded rectangle R_i corresponds to element L_i in L , calculated as the weighted average of the pixels fully, or partly, covered by R_i . Each pixel contributes in direct proportion to its area covered by R_i . The exact end-points of L need not lie on a corner of the boundary pixels B_1 and B_2 .

For each class k in the training set of V_k face views, we randomly generate $N_k = V_k \times N_{V_k}$ lattice lines (N_{V_k} lines per face view per class), $L_{i,k} \equiv (L_{i,k}^{(1)}, L_{i,k}^{(2)}, \dots, L_{i,k}^{(l)})$ for $i = 1, 2, \dots, N_k$. There are $M = \sum_{k=1}^K N_k$ such lattice lines for K face classes. The set of lattice lines for all K face classes is given by:

$$\Psi = \bigcup_{k=1}^K \bigcup_{i=1}^{N_k} L_{i,k}$$

The distance $D(L_{i,k}, L_{m,n})$ between two lattice lines $L_{i,k}$ and $L_{m,n}$ can be defined as

$$D(L_{i,k}, L_{m,n}) = \sum_{q=1}^l (L_{i,k}^{(q)} - (L_{m,n}^{(q)} + \Delta))^2 \quad (1)$$

where $\Delta = \mu(L_{i,k}) - \mu(L_{m,n})$ and $\mu(L_{s,t}) = \sum_{q=0}^l L_{s,t}^{(q)} / l$. The value of Δ has the effect of shifting the two lines towards the same average value, making the distance measure invariant to illumination intensity (not to be confused with the illumination direction).

Given an unseen test lattice line L_j where generally $L_j \notin \Psi$, we define $L_{j,*}$ such that $D(L_j, L_{j,*})$ is the minimum, where $L_{j,*} \in \Psi$. The Nearest-Neighbour Classifier (NNC) maps L_j to the class k to which $L_{j,*}$ belongs. That is, $\text{NNC}(L_j) = k$. We write D_j for $D(L_j, L_{j,*})$.

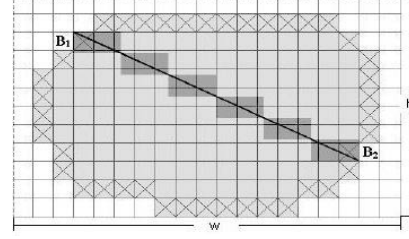


Fig. 1. A definition of an image lattice line for $l = 6$ (Presented by O.de Vel and S.Aeberhard). Crossed squares $\in \beta$.

Assume that there be N test lines L_j , $j = 1, 2, \dots, N$, and for each line we have $L_{j,*}$ and D_j .

Let $D_{max} = k_1 \times \max_{1 \leq j \leq N} \{D_j\}$ (for some value k_1 , where $0 < k_1 \leq 1$) and $D_{min} = k_1 \times \min_{1 \leq j \leq N} \{D_j\}$. We define the cumulative error for line L_j , $e_j = (\sum_{q=1}^l (|L_{j,*}^{(q+1)} - L_{j,*}^{(q)}|)) / (l - 1)$ for $q = 1, 2, \dots, l - 1$ and the maximum cumulative error statistic, $e_{max} = \max_{1 \leq j \leq N} \{e_j\}$.

Define the measure of confidence that $\text{NNC}(L_j)$ is correct, C_j :

$$C_j = \begin{cases} 0, & \text{if } D_j > D_{max} \\ \frac{D_{max} - D_j}{D_{max} - D_{min}} w_1 [\frac{e_j}{e_{max}}]^{p_2}, & \text{otherwise} \end{cases}$$

where p_1, p_2, w_1 and $w_2 \in R^+$. The variables p_1 and p_2 control the shape of the confidence function, whereas w_1 and w_2 are the weight magnitudes of the distance and cumulative error statistic components, respectively.

To classify a face F for which we know its boundary pixel set β , we randomly select N lattice lines L_j ($j = 1, 2, \dots, N$). For each face class F_k ($k = 1, 2, \dots, K$), let $T_k = \sum_{j=1}^N C_j$, such that $\text{NNC}(L_j) = F_k$. We assign F to class g when T_g is the maximum. That is, if $T_g = \max_{1 \leq j \leq N} \{T C_j\}$, then $F \in F_g$ ($g = 1, 2, \dots, K$).

3. OUR APPROACH

In this section, we will indicate a theoretic drawback of equation (1) and re-define equation (1). Then, a deduction will be given to prove the invariance of our approach to illumination intensity. Finally, we suggest a distributed computing algorithm to solve the huge computation problem.

3.1. Calculating distance using LSV feature vector

In equation (1), the illumination intensity will greatly influence the final result of $D(L_{i,k}, L_{m,n})$ because the gray-level value is directly used. Although the value of Δ has been introduced to shift the two lines towards the same average value, great influence inevitably happens for the theoretic drawback. To further decrease the effect of illumination intensity, we introduce the line-based singular value feature vector.

For each class k in the training set of V_k face views, we randomly generate $N_k = V_k \times N_{V_k}$ lattice lines (N_{V_k} lines

per face view per class), $L_{i,k} \equiv (L_{i,k}^{(1)}, L_{i,k}^{(2)}, \dots, L_{i,k}^{(l)})$ for $i = 1, 2, \dots, N_k$.

Let $l = s \times s$ ($s > 0$), then

$$L_{i,k} \equiv \underbrace{(L_{i,k}^{(1)}, \dots, L_{i,k}^{(s)})}_1 \underbrace{(L_{i,k}^{(s+1)}, \dots, L_{i,k}^{(2 \times s)})}_2 \dots \underbrace{(L_{i,k}^{(s^2-s+1)}, \dots, L_{i,k}^{(s \times s)})}_s$$

We define a $s \times s$ matrix $M_{L_{i,k}}$ as

$$M_{L_{i,k}} = \begin{pmatrix} L_{i,k}^{(1)} & L_{i,k}^{(2)} & \dots & L_{i,k}^{(s)} \\ L_{i,k}^{(s+1)} & L_{i,k}^{(s+2)} & \dots & L_{i,k}^{(2 \times s)} \\ \dots & \dots & \dots & \dots \\ L_{i,k}^{(s^2-s+1)} & L_{i,k}^{(s^2-s+2)} & \dots & L_{i,k}^{(s \times s)} \end{pmatrix}$$

We will introduce theorem 1 [7] to show an important property of any matrix. Denote the field of quaternion by H , all the $M \times N$ matrix quaternion valued by $H^{M \times N}$. We firstly give the following singular value decomposition of quaternion matrix.

Theorem 1 For any matrix $S \in H^{M \times N}$ with the rank r , there exist two unitary quaternion matrices U and V such that

$$S = U \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} V^\diamond$$

where $\Sigma_r = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_r)$ is a real diagonal matrix and has r non-null entries λ_i ($1 \leq i \leq r$) on its diagonal (i.e. singular values of S); the sign “ \diamond ” is the conjugate transposition operator. $U \in H^{M \times N}$ and $V \in H^{M \times N}$ contain respectively the left and right quaternion singular vectors of S .

The proof of the above theorem is similar to the proposition in [8]. According to theorem 1, the $s \times s$ matrix $M_{L_{i,k}}$ can be denoted as

$$M_{L_{i,k}} = U' \Sigma' V^T$$

where $\Sigma' = \text{diag}(\lambda'_1, \lambda'_2, \dots, \lambda'_r, 0, \dots, 0)$ is a real diagonal matrix and has r non-null entries λ'_i ($1 \leq i \leq r$) on its diagonal.

Then we define a $(s \times 1)$ column vector $LSV_{(L_{i,k})}$ as

$$LSV_{(L_{i,k})} = (\lambda'_1, \lambda'_2, \dots, \lambda'_r, 0, \dots, 0)^T$$

Finally, the distance $D(L_{i,k}, L_{m,n})$ between two lattice lines $L_{i,k}$ and $L_{m,n}$ can be re-defined as

$$D(L_{i,k}, L_{m,n}) = \sum_{q=1}^s (LSV_{(L_{i,k})}^{(q)} - LSV_{(L_{m,n})}^{(q)})^2 \quad (2)$$

Some properties (stable, invariant to rotation, translation and proportion transformation) of the LSV feature vector are discussed in [7].

A disadvantage of our approach is that it may require additional computation to calculate the LSV feature vector. Its super performance, however, seems to compensate the additional computational effort (See Figure 2 (b)).

3.2. Invariance to changes of illumination intensity

We will discuss why our approach is invariant to changes of illumination intensity here. Our aim is to prove that the LSV feature vector changes in proportion with the linearly fluctuation of illumination intensity.

Assume that the gray-level value matrix of a original image be denoted as A . Linear changing in illumination means to multiply A by a non-zero real number α , then we get a image matrix αA . Of course $\text{rank}(A) = \text{rank}(\alpha A)$.

Let $\text{rank}(A) = k$, and the singular value of A and αA are $\lambda_1, \lambda_2, \dots, \lambda_k$ and $\sigma_1, \sigma_2, \dots, \sigma_k$ respectively. Then the eigenquation of $(\alpha A)(\alpha A)^T$ can be denoted as

$$|(\alpha A)(\alpha A)^T - \sigma^2 I| = 0$$

such that

$$\left| AA^T - \frac{1}{\alpha^2} \sigma^2 I \right| = 0$$

Then we can get that

$$(\lambda_1, \lambda_2, \dots, \lambda_k)^T = \frac{1}{|\alpha|} (\sigma_1, \sigma_2, \dots, \sigma_k)^T \quad (3)$$

Equation (3) shows that an equal of linear changing in illumination is to multiply the singular value feature vector by a factor α . We can simply erase the influence of illumination intensity changes by normalizing the LSV feature vector, and the normalization of LSV feature vector will not result in the information loss for recognition.

3.3. Distributed computing algorithm

Grid computing is a computational concept based on an infrastructure that integrates and collaborates the use of high end computers, networks, databases and scientific instruments owned and managed by organizations [9]. We build a distributed computing model in a grid environment for parallel and distributed computing. We use grid computing nodes (GCNs) to undertake the computation tasks.

To classify a face X for which we have detected its boundary pixel set β , our distributed recognition algorithm is described in Table 1 according to the line-based face recognition algorithm. If we have enough GCNs, the computation task mentioned in step 3 can be further divided into N_k sub-tasks.

4. RESULTS AND ANALYSIS

We use a face database established by ourselves to evaluate the performance of our algorithm. Pictures of 35 persons are taken by a standard camera (6 pictures per person) under different environments of illumination intensity (weak, medium and strong). We select 3 views of each person for training, and the other 3 views (in weak, medium, and strong illumination intensity respectively) is used to test. Figure 2 (a) shows some samples of our database.

Table 1. THE DISTRIBUTED RECOGNITION ALGORITHM

```

RecognizeOnGrid( $X$ )
/*  $X$  represents the input face image*/
begin
  Step 1: for each face class  $F_k$  ( $k = 1, 2, \dots, K$ ), read
     $N_k = V_k \times N_{V_k}$  lattice lines ( $N_{V_k}$  lines per face
    view per class,  $V_k$  face views in  $F_k$ ) from face
    database;
  LineFinished = 0;
  While (LineFinished <  $N$ )
  begin
    Step 2: select a lattice lines  $L_j$  ( $j = 1, \dots, N$ )
      randomly in  $X$ , and scale the line to length
       $l$  ( $l = s \times s$ );
    LineFinished = LineFinished + 1;
    if (existing a spare GCN)
      Step 3: for a test lattice line  $L_j$ , assign a spare
        GCN to map  $L_j$  to a face class  $F_k$  using
        NNC classifier;
    else
      Wait a second;
    end
  Step 4: calculate a  $TC_g = \max_{1 \leq j \leq N} \{TC_j\}$ ;
  Step 5: return  $X \in F_g$  ( $g = 1, 2, \dots, K$ );
end

```

We have compared our results with benchmark results obtained by other workers (e.g. PCA, LDA) on the same, but individually processed, face database.

As illustrated in Figure 2 (b) and Figure 3, our recognition results are better than the benchmarks on the database, changes of illumination intensity influence little on recognition. Furthermore, our execution times can be shortened by increasing number of GCNs.

5. CONCLUSIONS

We have described an improved approach to the line-based face recognition algorithm. For changes in illumination intensity, the LSV feature is used instead of the gray-level value. For training and recognition, a distributed algorithm based on grid computing is used.

The LSV feature vector is invariant to the illumination intensity. This super performance seems to compensate the additional computational effort. Also, a distributed computing algorithm using grid computing has been given.

The main drawback of our technique lies in the huge computation. We are currently investigating modifications to the algorithm that will account for the decrease of computation.

6. REFERENCES

[1] O.de Vel and S.Aeberhard., "Line-based face recognition under

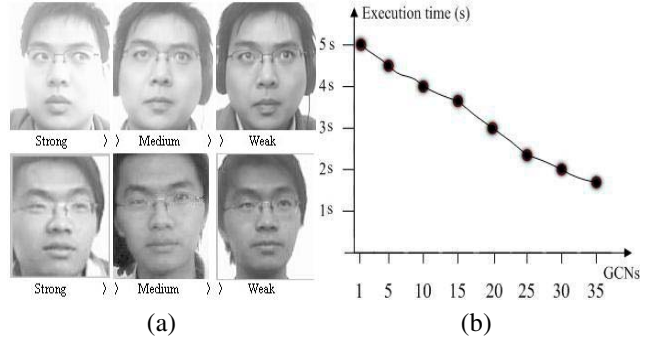


Fig. 2. (a) Samples of our face database. (b) The relation between the execution times and the numbers of GCNs.

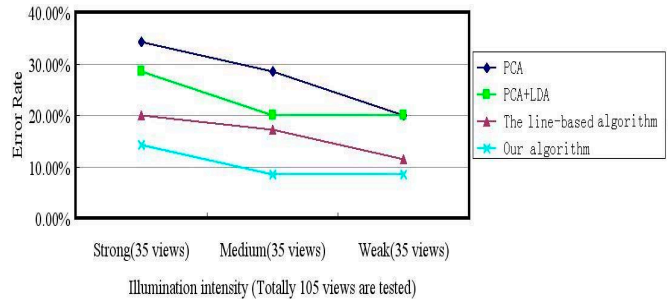


Fig. 3. For the variable illumination intensity, our algorithm take a stable performance on error rate (Totally 105 pictures are tested).

varying pose", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp. 1081-1088, 1999

[2] O.de Vel and S.Aeberhard., "View-based object recognition using image line", *Proceedings 14th International Conference on Digital Object Identifier*, vol. 2, pp. 1198-1200, 1998.

[3] O.de Vel and S.Aeberhard., "Face recognition using multiple image view line segments", *Proceedings 14th International Conference on Pattern Recognition*, Brisbane, Australia, 1998.

[4] S.K. Singh, M.Vatsa, R.Singh, D.S. Chauhan, "A comparison of face recognition algorithms neural network based & line based approaches", *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 6, pp. 6-9, 2002

[5] Zhao WY, Chellappa R, Phillips PJ, Rosenfeld.A, "Face recognition: A literature survey". *ACM Computing Surveys*, 2003, 35(4):399-458.

[6] M.Lades, J.Vorbruggen, J.Buhmann, J.Lange, C.V.D. Malburg, and R.Wurtz, "Distortion invariant object recognition in the dynamic link architecture", *IEEE Trans. Comput.*, vol. 42, no. 3, pp. 300-311, 1993.

[7] N.L. Bihan, and J. Mars, "Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing", *Signal processing*, Vol. 84, pp. 1177-1199, 2004.

[8] Z.Q. Hong, "Algebraic feature extraction of image for recognition", *Pattern Recognition*, Vol. 24, No. 3, pp. 211-219, 1991.

[9] D.D. Kouvatso and I.M. Mkwawa, "Multicast communication in grid computing networks with background traffic", *Proceedings: Software*, vol. 150, pp. 257-264, 2003.