A novel pose and illumination robust face recognition with a single training image per person algorithm

Junbao Li (李君宝)[1] and Jeng-Shyang Pan (潘正祥)[2]

[1]*Department of Automatic Test and Control, Harbin Institute of Technology, Harbin 150001*

[2]*Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung*

Received October 8, 2007

In the real-world application of face recognition system, owing to the difficulties of collecting samples or storage space of systems, only one sample image per person is stored in the system, which is so-called one sample per person problem. Moreover, pose and illumination have impact on recognition performance. We propose a novel pose and illumination robust algorithm for face recognition with a single training image per person to solve the above limitations. Experimental results show that the proposed algorithm is an efficient and practical approach for face recognition.

OCIS codes: 100.5010, 100.5760, 100.0100, 100.2960.

Face recognition has received more attention from the industrial communities in the recent years owing to its potential applications in information security, law enforcement and surveillance, smart cards, access control, and so on[1−3]. In many practical applications, because of the difficulties of collecting samples or storage space of systems, only one sample image per person is stored in the system, so the research of face recognition from one sample per person, owing to its own advantages (easy collecting of samples, less storage and computational cost), has been a sub-research topic in the face recognition area. The traditional method such as Fisherface[4] fails when each person just has one training face sample available because of nonexistence of the intra-class scatter. Recently, researchers have proposed many algorithms, such as (PC)^{2}A[5], E(PC)^{2}A[6], and singular value decomposition (SVD) perturbation[7], for face recognition with one training image per person. But these algorithms still endure some problems. For example, the procedure of E(PC)^{2}A is divided into two stages: 1) constructing a new image by combining the first-order and second-order projected images and the original image; 2) performing principal component analysis (PCA) on the newly-combined training images. In the second stage, the combined image matrix should be mapped into a one-dimensional (1D) vector in advance in order to perform PCA. This causes the high storage and computational cost. In order to enhance the practicability of the face recognition system, we propose a novel algorithm called 2D(PC)^{2}A for face recognition with one training image per person in this letter. 2D(PC)^{2}A performs PCA on the set of combined training images directly without mapping the image matrix to 1D vector. Thus 2D(PC)^{2}A can directly extract feature matrix from the original image matrix. This leads to that much less time is required for training and feature extraction. Further, experiments implemented on two popular databases show that the recognition performance of 2D(PC)^{2}A is better than that of classical E(PC)^{2}A.

The procedure of 2D(PC)^{2}A can be divided into three stages: 1) creating the combined image from the original image $I(m,n)$ with $M \times N$ pixels ($I(m,n) \in [0,1]$, $m \in [1,M]$, $n \in [1,N]$); 2) performing two-dimensional PCA (2DPCA) on the combined images; 3) classifying a new face based on assembled matrix distance (AMD). The detailed procedure is described as follows.

Step 1: creating the combined image. In order to effectively recognize faces with only one example image per class, we derive a combined image from the original image by the first-order and second-order projection. The first-order projected image $P_1(m,n)$ and second-order projected image $P_2(m,n)$ are created as

$$P_1(m,n) = \frac{V_1(m)H_1(n)}{MN\overline{I}}, \tag{1}$$

$$P_2(m,n) = \frac{V_2(m)H_2(n)}{MN\overline{J}}, \tag{2}$$

where $V_1(m) = \frac{1}{N}\sum_{p=1}^{N} I(m,p)$ and $H_1(n) = \frac{1}{M}\sum_{q=1}^{M} I(q,n)$, and $\overline{I}$ is the mean value of $I(m,n)$; $V_2(m) = \frac{1}{N}\sum_{n=1}^{N} J(m,n)$ and $H_2(n) = \frac{1}{M}\sum_{m=1}^{M} J(m,n)$, and $J(m,n) = I(m,n)^2$ and $\overline{J}$ is the mean value of $J(m,n)$. Then the combined image can be created as

$$I_p(m,n) = \frac{I(m,n) + \alpha P_1(m,n) + \beta P_2(m,n)}{1+\alpha+\beta}. \tag{3}$$

Step 2: performing 2DPCA. Instead of performing PCA on the set of combined images, 2D(PC)^{2}A performs 2DPCA on the image matrix directly rather than 1D vectors for covariance matrix estimation, thus it is claimed to be more computationally cheap and more suitable for small sample size problem. Let the combined image be I_{pj} $(j = 1, 2, \cdots, C)$ and the average image of all training samples be $\overline{I_p}$, then the image convariance matrix S_T can be evaluated as

$$S_T = \frac{1}{C}\sum_{j=1}^{C} \left(I_{pj} - \overline{I_p}\right)^T \left(I_{pj} - \overline{I_p}\right). \tag{4}$$

1671-7694/2008/040255-03 © 2008 Chinese Optics Letters

Then, a set of optimal projection axis of 2DPCA $\{w_1, w_2, \cdots, w_d\}$, which are then used for feature extraction, can be obtained by maximizing the image scatter criterion:

$$J(W) = W^T S_T W. \tag{5}$$

The low-dimensional feature matrix Y of a combined image matrix I_p can be obtained as

$$Y = I_p W_{\mathrm{opt}}, \tag{6}$$

where $W_{\mathrm{opt}} = \{w_1, w_2, \cdots, w_d\}$. In Eq. (6) the dimension of 2DPCA projector W_{opt} is $N \times d$, and the dimension of 2DPCA feature matrix Y is $M \times d$.

Step 3: implementing AMD for classification. After the feature matrices are extracted from the original images based on 2D(PC)^{2}A in Steps 1 and 2, the nearest neighbor criterion is applied to classification based on the distance between two feature matrices. Unlike E(PC)^{2}A approach to produce a feature vector, 2D(PC)^{2}A directly extracts a feature matrix from an original image matrix. So we apply AMD metric to calculate the distance between two feature matrices. Given two feature matrices $A = (a_{ij})_{M\times d}$ and $B = (b_{ij})_{M\times d}$, the AMD is obtained as

$$d_{\mathrm{AMD}}(A, B) = \left(\sum_{j=1}^{d} \left(\sum_{i=1}^{M} (a_{ij} - b_{ij})^2 \right)^{(1/2)p} \right)^{1/p}. \tag{7}$$

After calculating the AMD between the feature matrices of the test sample and the training sample, we apply the nearest neighbor criterion to classification based on AMD. Experiments were implemented on ORL[8], YALE[9] and UMIST[10] databases to evaluate the proposed algorithm.

ORL face database, developed at the Olivetti Research Laboratory, Cambridge, UK, is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time and facial expression. Some image examples are shown in Fig. 1.

YALE face database was constructed at the YALE Center for Computational Vision and Control, New Haven, USA. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting conditions (left-light, center-light, and right-light), and different facial expressions (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Some image examples are shown in Fig. 2.

Fig. 1. Examples from ORL face database (the Olivetti Research Laboratory, UK).

Fig. 2. Examples from YALE face database (the YALE Center for Computational Vision and Control, USA).

Fig. 3. Examples from UMIST face database (the University of Manchester Institute of Science and Technology, UK).

Table 1. Deterministic Training and Test Set on ORL Face Database

	Training Set	Test Set
ORL_A	1#	2#,3#,4#,5#,6#,7#,8#,9#,10#
ORL_B	2#	1#,3#,4#,5#,6#,7#,8#,9#,10#
ORL_C	3#	1#,2#,4#,5#,6#,7#,8#,9#,10#
ORL_D	4#	1#,2#,3#,5#,6#,7#,8#,9#,10#
ORL_E	5#	1#,2#,3#,4#,6#,7#,8#,9#,10#

1# denotes the first image of each person, and other images are marked with the same way.

Table 2. Deterministic Training and Test Set on YALE Face Database

	Training Set	Test Set
YALE_a	1#	2#,3#,4#,5#,6#,7#,8#,9#,10#,11#
YALE_b	2#	1#,3#,4#,5#,6#,7#,8#,9#,10#,11#
YALE_c	3#	1#,2#,4#,5#,6#,7#,8#,9#,10#,11#
YALE_d	4#	1#,2#,3#,5#,6#,7#,8#,9#,10#,11#
YALE_e	5#	1#,2#,3#,4#,6#,7#,8#,9#,10#,11#

UMIST face database, established by the University of Manchester Institute of Science and Technology, Manchester, UK, consists of 564 images of 20 people. Each covers a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. Each subject exists in their own directory labelled 1a, 1b, $\cdots$, 1t and images are numbered sequentially as they were taken. Some image examples are shown in Fig. 3.

We implemented experiments on ORL and YALE with two manners. For the deterministic manner, the training set and the testing set are constructed as shown in Tables 1 and 2. Here our goal is to have a good look at the performance of specific partition of the database, thus we can see how much the influence of recognition rate under the different pose, illumination and expression (PIE). For the random manner, from ORL face database, we randomly select one image from each subject, and the rest images are used to test the performance. Only one image of each person randomly selected from YALE database are used to construct the training set, and the rest images of each person are used to test the performance of the algorithms. Moreover, we implemented experiments on UMIST face database on a random manner.

It is worthy to emphasize the following points. 1) We run experiments for 10 times, and the average rate is used to evaluate the classification performance. 2) The experiments are implemented on a Pentium 3.0 GHz computer with 512-MB RAM and programmed in the MATLAB platform (Version 6.5). 3) To reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels. Similarly, the images from YALE databases are cropped to the size of 100×100 pixels, finally a subimage procedure crops the face image to the size of 112×92 to extract the facial region on UMIST face database.

We also implement other popular methods such as PCA, (PC)^{2}A, E(PC)^{2}A, and SVD perturbation for face recognition with single training sample per person. In our experiments, we select $\alpha = 0.125$ and $\beta = 0.05$ for 2D(PC)^{2}A and E(PC)^{2}A. As shown in Tables 3 – 7, the proposed algorithm gives a highest recognition rate compared with other popular methods. Moreover, since 2D(PC)^{2}A deals with matrix directly instead of mapping into 1D vector as E(PC)^{2}A or (PC)^{2}A, it is apparent that 2D(PC)^{2}A is more efficient than E(PC)^{2}A or (PC)^{2}A. So we say that 2D(PC)^{2}A method is an efficient and practical approach for face recognition.

Table 3. Recognition Performance on ORL Database in Random Manner

Algorithm	PCA	2DPCA	(PC)^{2}A
Recognition Rate	0.54	0.54	0.56
Algorithm	E(PC)^{2}A	SVD	2D(PC)^{2}A
Recognition Rate	0.57	0.55	0.60

Table 4. Recognition Performance on YALE Database in Random Manner

Algorithm	PCA	2DPCA	(PC)^{2}A
Recognition Rate	0.54	0.56	0.55
Algorithm	E(PC)^{2}A	SVD	2D(PC)^{2}A
Recognition Rate	0.56	0.54	0.61

Table 5. Recognition Performance on ORL Database in Deterministic Manner

Algorithm	ORL_A	ORL_B	ORL_C	ORL_D	ORL_E
PCA	0.55	0.55	0.55	0.57	0.57
2DPCA	0.54	0.55	0.56	0.56	0.57
(PC)^{2}A	0.57	0.58	0.57	0.59	0.61
E(PC)^{2}A	0.58	0.59	0.58	0.60	0.62
SVD	0.56	0.59	0.59	0.59	0.60
2D(PC)^{2}A	0.61	0.62	0.61	0.63	0.64

Table 6. Recognition Performance on YALE Database in Deterministic Manner

Algorithm	YALE_a	YALE_b	YALE_c	YALE_d	YALE_e
PCA	0.55	0.56	0.54	0.57	0.56
2DPCA	0.56	0.57	0.55	0.58	0.59
(PC)^{2}A	0.55	0.57	0.54	0.57	0.58
E(PC)^{2}A	0.57	0.58	0.56	0.59	0.58
SVD	0.55	0.56	0.55	0.58	0.58
2D(PC)^{2}A	0.62	0.63	0.61	0.64	0.63

Table 7. Recognition Performance on UMIST Database in Random Manner

Algorithm	PCA	2DPCA	(PC)^{2}A
Recognition Rate	0.56	0.57	0.59
Algorithm	E(PC)^{2}A	SVD	2D(PC)^{2}A
Recognition Rate	0.60	0.59	0.65

Although the proposed algorithm gives a highest recognition rate compared with other popular methods, the highest recognition rate (only about 0.60) is still not so high owing to PIE problem of face recognition. So in the future work, we will pay attention to solve the PIE problem to enhance the whole recognition rate of the algorithm. Some essential questions to be answered in the future are included here. 1) Are there other methods of choosing α and β in the practical application? 2) In experiments, the parameter p for AMD is chosen with the experiments. Are there any other alternative methods to choose this parameter? 3) 2D(PC)^{2}A gives a higher recognition accuracy, but the recognition rate is not so high. How to increase the recognition performance of the algorithm is a key problem.

J. Li's e-mail address is junbaolihit@hotmail.com.

References

1. D. Zhou, X. Yang, and N. Peng, Chin. Opt. Lett. **3,** 330 (2005).
2. X. Chen, Z. Jing, S. Sun, and G. Xiao, Chin. Opt. Lett. **2,** 694 (2004).
3. G. Liu, X. Lü, and G. Huang, Chin. Opt. Lett. **5,** 636 (2007).
4. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, IEEE Trans. Pattern Analysis and Machine Intelligence **19,** 711 (1997).
5. J. Wu and Z.-H. Zhou, Pattern Recogn. Lett. **23,** 1711 (2002).
6. S. Chen, D. Zhang, and Z.-H. Zhou, Pattern Recogn. Lett. **25,** 1173 (2004).
7. D. Zhang, S. Chen, and Z.-H. Zhou, Appl. Math. Comput. **163,** 895 (2005).
8. F. S. Samaria and A. C. Harter, in *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision* 138 (1994).
9. http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
10. http://images.ee.umist.ac.uk/danny/database.html.