

RESEARCH

Open Access



Cross-pose face recognition by integrating regression iteration and interactive subspace

Kefeng Li* and Quanzhen Huang

Abstract

At present, the pose change of the face test sample is the main reason that affects the accuracy of face recognition, and the design of cross-pose recognition algorithm is a technical problem to be solved. In this paper, a cross-pose face recognition algorithm which integrates the regression iterative method and the interactive subspace method was proposed, and through the regression iteration, the target function converges rapidly and important characteristics of the sample were extracted. Then, the posture of cross-pose face image was estimated, and finally, the interactive subspace method was applied to judge the similarity of the human face. The experimental results of FERET face database ($\pm 45^\circ$ and $\pm 90^\circ$ posture) and MIT-CBCL face database (N-fold cross-validation) showed that the proposed RIM-ISM algorithm had a higher recognition rate and robustness, and it could effectively solve the difficulty of cross-pose face recognition.

Keywords: Regression iterative method (RIM), Cross-pose face recognition, Pose estimation, Interactive subspace method (ISM), N-fold cross-validation

1 Introduction

The cross-pose face recognition problem is caused by texture changes brought by the rotation transformation of the face. When face pose change is bigger, and within-cluster variation is greater than inter-cluster variation, the recognition rate will fell sharply. Therefore, cross-pose face recognition is an important research topic in recent years.

In order to solve the problem of cross-pose recognition, there are two main research methods: the algorithms based on pose correction and feature-based algorithms. Pose correction method rotates the face image and fits it into a new image, and this synthesis method includes two types of face models based on 3D and 2D. Y. Taigman et al. proposed DeepFace [1]. First of all, 3D face alignment was used for face pose correction, and then face frontal was processed. Later, it was then fed into the convolutional neural network for feature extraction, using the classifier for face authentication. Masi et al. proposed pose-aware CNN models (PAMs) method

to deal with pose changes for large pose changes [2]. The method uses a convolutional neural network to learn a specific multi-pose model and converts the multi-pose image to the learned model. Zhu et al. designed a new multi-view sensor, which can separate pose and sample information by using different neurons in the network and reconstruct various pose images of a single 2D image [3]. M. Kan et al. designed a stacked progressive auto-encoders (SPA-E) neural network model to achieve nonlinear modeling of posture changes in a small-scale data [4]. J. Yim et al. designed a deep neural network based on multi-task learning, the network can achieve any pose and any illumination face rotated to the designated face pose and normal illumination. Feature-based methods are to extract face features which are robust to pose for face recognition [5]. Y. Sun proposed DeepID network structure, which considered local and global characteristics simultaneously, and introduced a larger data set for model training [6]. Florian Schroff et al. proposed FaceNet framework to directly learn the coding method from image to Euclidean space, and then conduct face recognition, face verification and face clustering on basis of this coding [7]. Omkar M.

* Correspondence: kefengli922@126.com

School of Electrical Information Engineering, Henan University of Engineering, Zhengzhou, China

Parkhi applied the deep network VGG to face recognition task. This network structure is implemented by a 16-layer convolutional neural network and a three-layer fully connected layer, and it has been tested in LFW and YouTube Faces database and achieved a better recognition effect [8]. Recently, cross-pose recognition methods based on features such as scale-invariant feature transform (SIFT) [9], LBP [10], and Gabor [11] have achieved good recognition effects under controlled conditions and partially unconstrained conditions.

Based on the study of these two kinds of methods, and through combining regression iterative method with interactive subspace method, this study designed a RIM-ISM cross-pose recognition algorithm: first of all, the regression iteration method was adopted to regress cross-pose face shape to the location close to the real shape and detect important characteristics of the sample. Secondly, the deflection angle of the 2D face image corresponding to the 3D rectangular coordinate system was calculated in the 2D plane and the pose correction was carried out. Finally, the similarity of cross-pose facial image was measured by using the interactive subspace method. In order to verify the effectiveness of RIM-ISM algorithm, large-scale experiments were conducted in the FERET database and the MIT-CBCL database. Compared with other algorithms, the experimental results show that the proposed RIM-ISM cross-pose face recognition algorithm had great advantages.

2 Methodology

2.1 Extraction of important characteristics

Cross-pose face recognition is to recognize or identify faces of any pose in an image. The human face is a 3D structure, and the face pose can change in the three dimensions (X , Y , and Z), as shown in Fig. 1. Pitch = θ_x rotates around X -axis, yaw = θ_y rotates around Y -axis horizontally, and roll = θ_z rotates around Z -axis plane.

The position of key feature points changes due to the change of pose, and the regression iterative method was adopted in this paper to deal with the change. The basic principle of regression iterative method is to provide an initial shape for a given face image, and through constant iterations, the initial shape is regressed to the position which is close to or equal to that of the real shape, and then obtains a series of downward directions and the directions on the scale, so as to make the target function converge to the minimum value quickly through studying, which improves the running speed of the program.

Given a face image containing m pixels, $x \in R^{m \times 1}$, $d(x) \in R^p \times 1$ represent p important characteristics of the sample, and $h(\cdot)$ represents a nonlinear characteristic extraction function. Then, on the basis of a given initial shape x_0 , regression iterative method was used to regress

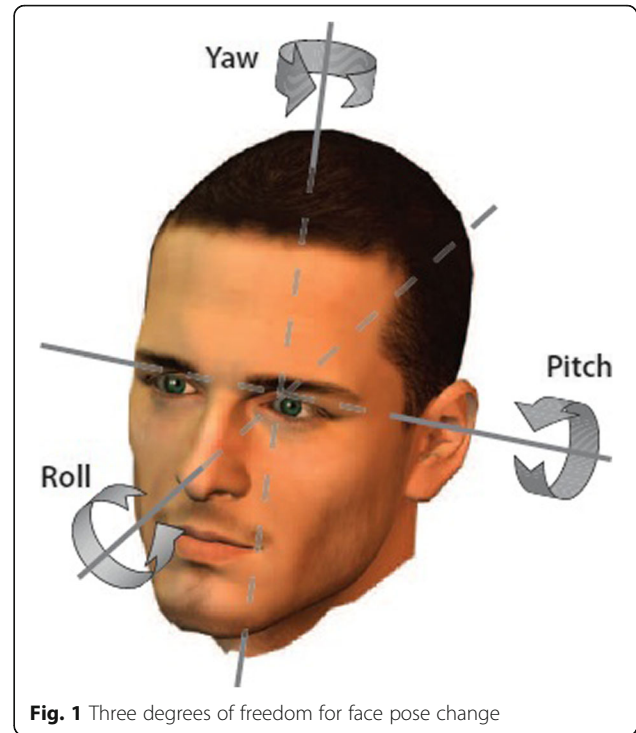


Fig. 1 Three degrees of freedom for face pose change

x_0 to the correct shape x_* of the face, that is, to solve Δx which minimized $f(x_0 + \Delta x)$ in Formula (1):

$$f(x_0 + \Delta x) = \|h(d(x_0 + \Delta x)) - \phi_*\|_2^2 \quad (1)$$

Where $\phi_* = h(d(x_*))$ is SIFT characteristics obtained from the real feature points of the face. The objective function was expanded by Taylor expansion as follows:

$$f(x_0 + \Delta x) \approx f(x_0) + J_f(x_0)^T \Delta x + \frac{1}{2} \Delta x H(x_0) \Delta x \quad (2)$$

Where, $J_f(x_0)$ is Jacobian matrix, $H(x_0)$ is Hessian matrix, and Formula (2) solved the derivative of Δx :

$$\Delta x = -H^{-1}(x_0) J_f(x_0) = -2H^{-1} J_h^T (\phi_{k-1} - \phi_*) \quad (3)$$

Where $\Delta x = x_k - x_{k-1}$, J_h is Jacobian matrix, and $\phi_{k-1} = h(d(x_{k-1}))$ the feature vector extracted from face feature point x_{k-1} . The regression iterative method was applied to solve the optimization problem of Formula (1). The optimal solution could be obtained by solving Formula (3), and Formula (3) could be rewritten as the regression iterative formula:

$$\Delta x = R_{k-1} \phi_{k-1} + b_{k-1} \quad (4)$$

Where R_{k-1} is the gradient optimization direction, b_{k-1} is the average deviation parameter, and R_{k-1} and b_{k-1} could be obtained directly through training and learning. The regression iterative method could be used to detect

the important characteristics of the sample in real time and extract the eyes, mouth, nose, and other important characteristic positions of the sample. At last, a total of 45 pairs of important characteristics were detected, as shown in Fig. 2.

2.2 Estimation of pose deflection

Face pose deflection estimation is to calculate the deflection angle $(\theta_x, \theta_y, \theta_z)$ of the 2D face image corresponding to the 3D rectangular coordinate system in the 2D plane, extract the important characteristics (eyes, nose, mouth, etc.) of the face image, and then determine the feature correspondence between the 3D model and the 2D model.

The multi-pose sample image was defined as I_Q , and $p_i(x_i, y_i) i = 1, 2, \dots, 45$ is an important characteristic of the sample detected by the regression iterative method, as shown in Fig. 2c–d. According to the perspective projection principle of the camera, the corresponding relationship between the key feature point P of the 3D model and the key feature point p of the current face image I_Q could be obtained as follows:

$$(p_i^T, P_i^T) = (x_i, y_i, X_i, Y_i, Z_i)^T \tag{5}$$

The mapping relation between important characteristics from the 2D model to the 3D model was defined as C_Q , thus:

$$C_Q = A_Q [R_Q t_Q] p = C_Q P \tag{6}$$

Where A_Q is the camera's internal parameter mapping matrix, t_Q is the translation mapping matrix, and R_Q is the rotation mapping matrix. R_Q contains the attitude information of the image to be recognized, and its mapping relation is

$$R_Q = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{7}$$

Formula (5), (6), and (7) can determine the pose of the face image I_Q to be recognized, that is, the deflection angle $(\theta_x, \theta_y, \theta_z)$ of the 2D face image in the 3D coordinate system. The rotation matrix can be expressed as

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{8}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{9}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The rotation mapping matrix R_Q can be expressed as

$$R_Q = R_z(\theta_z) R_y(\theta_y) R_x(\theta_x) \tag{11}$$

Formula (8) to (11) can be used to calculate the deflection angle $(\theta_x, \theta_y, \theta_z)$, respectively. The calculation formula is as follows:

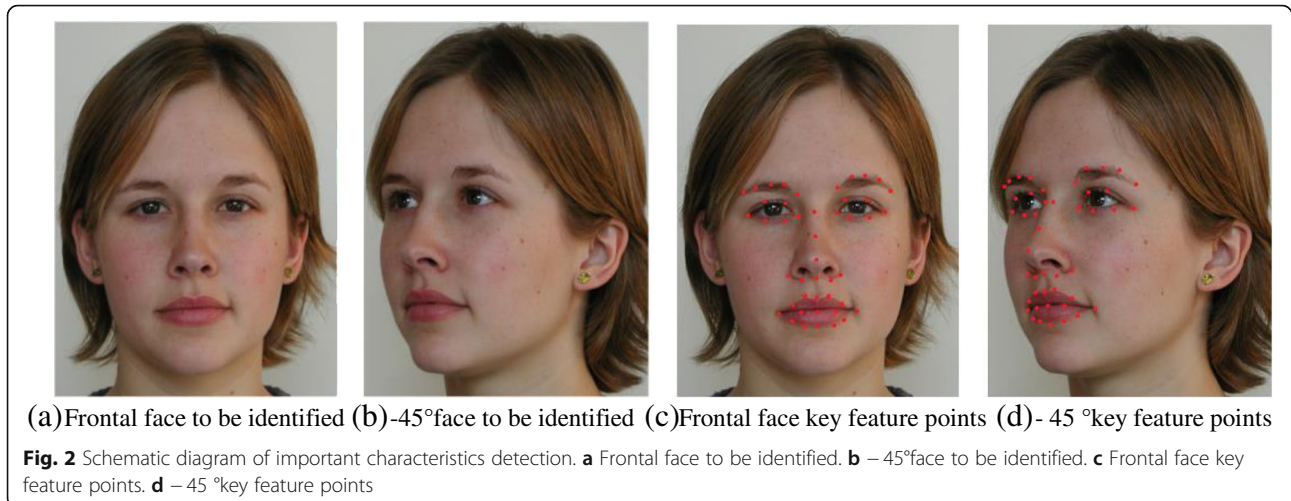
$$\theta_x = a \tan 2(R_{32}, R_{33}) \tag{12}$$

$$\theta_y = -\sin^{-1}(R_{31}) \tag{13}$$

$$\theta_z = a \tan 2\left(\frac{R_{21}}{\cos\beta}, \frac{R_{11}}{\cos\beta}\right) \tag{14}$$

2.3 Determination of face similarity

After the face deflection angle was estimated, the cross-pose face image was rotated to correct the pose, so



as to fit the approximate positive attitude image and establish the subspace G after pose correction, and vector $g \in G$. The reference subspace of frontal face was defined as D , vector $d \in D$; through calculating the similarity between two subspaces, it determined whether the two subspaces were the same category. Subspace method was usually used as the similarity criterion through computing two subspace cosine of the angle θ , namely,

$$\cos\theta = S(g) = \frac{1}{\|g\|^2} \sum_{n=1}^N (g, \phi_n)^2 \tag{15}$$

Where ϕ_n is the eigenvector of D , and N is the dimension of D . In order to avoid the interference brought by artificial selection of reference subspace, and improve the stability of the similarity between samples, this paper adopted the interactive subspace method (ISM) to measure the similarity between cross-pose face images: one subspace was used as the input subspace, and the other subspace was used as a reference subspace. Then the subspace method calculation was conducted for two times, and the two new spatial vectors were obtained. Later, the included angle between the two new vectors θ was calculated, and the principle was shown in Fig. 3.

The calculation formula is as follows:

$$\cos^2\theta = \sup_{d \in D, g \in G, \|d\| \neq 0, \|g\| \neq 0} \frac{|(d, g)|^2}{\|d\|^2 \|g\|^2} \tag{16}$$

Formula (16) has a local maximum value, so matrix P and Q were defined as the orthogonal projection matrix of subspace D and G , and the value of $\cos\theta$ could be obtained by calculating the maximum value λ_{\max} of the eigenvalue of matrix PQP , namely:

$$\cos\theta = \lambda_{\max} \tag{17}$$

Through using the interactive subspace method, the eigenvalue problem of calculating high-dimensional matrix PQP was transformed into calculating the eigenvalue of a low-dimensional matrix, and let the matrix X be

$$X = (x_{ij}) = \sum_{n=1}^N (\rho_i, \phi_n) (\phi_n, \rho_j) \tag{18}$$

Where ρ_j is the eigenvector of G , L is the dimensions of G ($L \leq N$), and the matrix X could be decomposed into

$$W^T X W = \Lambda \tag{19}$$

Where the diagonal value of the matrix Λ is the eigenvalue of X , and the maximum eigenvalue λ_{\max} was selected to evaluate the similarity between D and G :

$$S_{ISM}(D, G) = \lambda_{\max} \tag{20}$$

According to Formula (17), $0 \leq S_{ISM} \leq 1$, if S_{ISM} is closer to 1, the greater the similarity between the two samples will be, and the greater the possibility that the face image it represents is the same person. On the contrary, if S_{ISM} is closer to 0, the greater the difference between the two samples will be, and the face image it represents cannot be the same person.

3 Experiments

3.1 FERET database cross-pose recognition experiment

In order to verify the cross-pose face recognition performance of the proposed method, an experiment was conducted on FERET database, which was created by the FERET project of the U.S. military. It contains 14,051 cross-pose, multi-illumination grayscale face images. The

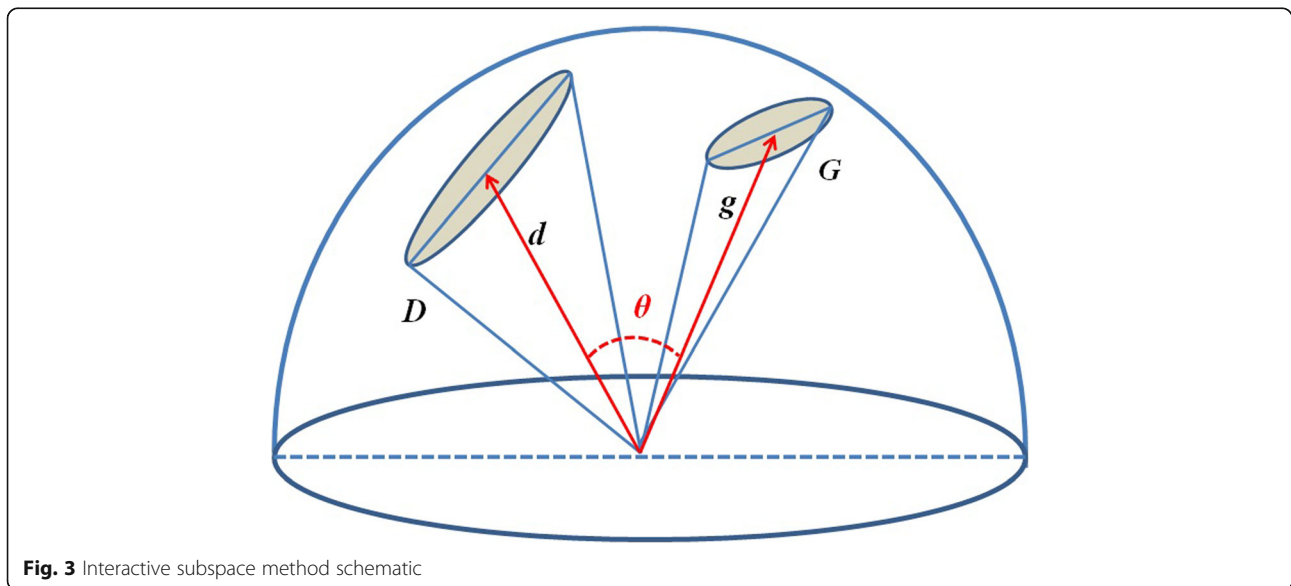


Fig. 3 Interactive subspace method schematic

database contains subsets for different orientations. Figure 4 shows an example of the images of FERET database posture subset.

Using the “leave-one-out” cross-validation method, the RIM-ISM method was compared with the three methods of DeepFace, SIFT, and Gabor, and a face image in the database was removed and used as test data, and the remaining images were used for training.

The experiment was repeated until all the images were tested again. Although the calculation was tedious, the sample utilization rate was the highest, so it was suitable for small samples.

We chose face image of $\pm 45^\circ$ and $\pm 90^\circ$ with a larger difficulty to test, and the experimental data were shown in Figs. 5 and 6.

3.2 N-fold cross-validation

In order to test the robustness and recognition efficiency of the RIM-ISM algorithm, N-fold cross-validation was performed in MIT-CBCL face database. The N-fold cross-validation randomly divided the sample data into N parts and selected one of them as the test sample set and N-1 parts as the training sample set. The cross-validation was repeated for N times, and each sub-sample was verified once, and the average result of N times was the estimation of the final single sample. The advantage of this test method is that it can randomly select generated sub-samples repeatedly for training and verification, so as to verify the robustness of face recognition algorithm in the face of random samples. At this time, the obtained average recognition rate and average recognition time had a more scientific statistical significance [12].

MIT-CBCL face database consisted of 10 people, and 200 images were collected from each of them, so there are a total of 2000 images. The sample image contained a large range of continuous pose changes, lighting and expression changes, in which the pose changes included horizontal pose changes and pitch angle changes. Figure 7 shows an example of MIT-CBCL face database pose change image.

The MIT-CBCL database was randomly divided into 10 parts, and each part contained 1 pose change image of each person. Therefore, each group was composed of 10 images, and each image corresponded to a different person. A 10-fold cross-validation test was conducted. In each experimental operation, 9 groups (90 images in total) were used as the training samples, and 1 group (10 images in total) was reserved as the test samples. The experimental data was shown in Table 1, showing the efficiency comparison of the RIM-ISM method with the DeepFace, SIFT, and Gabor methods.

4 Results and discussion

The data in Figs. 5 and 6 showed that when the samples' characteristic dimensions were relatively low, the RIM-ISM algorithm is close to the recognition rate of DeepFace algorithm. With the increase of the samples' characteristic dimensions, the RIM-ISM algorithm had a higher accuracy than DeepFace, SIFT, and Gabor method. In the experiment of $\pm 45^\circ$ pose change, the recognition rate was up to 97.1%, and in the experiment of $\pm 90^\circ$ pose change, the recognition rate was up to 79.3%, which were higher than the performance of the other three methods in the same condition. The recognition rate in Fig. 5 was significantly higher than that in Fig. 6, indicating that the difficulty of face recognition increased with the increase of pose deflection angle. Since the pose of the test samples was changed from ± 45 to $\pm 90^\circ$, the extraction of key feature points were insufficient, and the 45 pairs of feature points were reduced to 19 pairs, resulting in the calculation deviation of the three deflection angles in face pose deflection estimation. When using ISM to determine the similarity between cross-pose face images, the accuracy was greatly reduced.

As can be seen from the data in Table 1, for random multi-pose face samples, the RIM-ISM method proposed in this paper could achieve an average recognition rate of 97.52% in the 10-fold cross-validation test, which was higher than DeepFace, SIFT, and Gabor method. In this experiment, the average recognition time of a single test sample was 36.28 ms, which was far lower than that of



Fig. 4 Images of FERET database pose subset

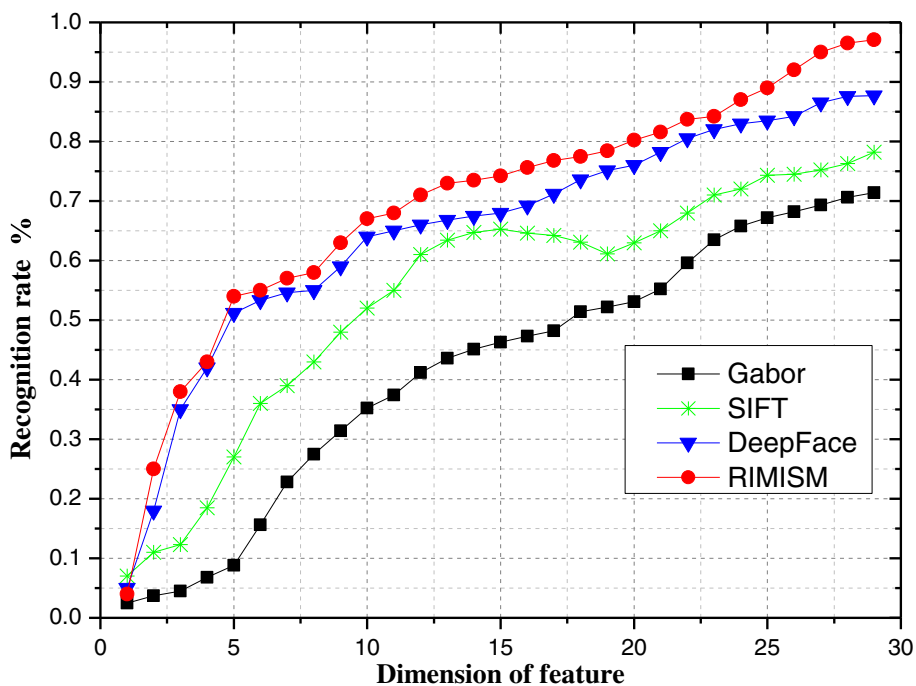


Fig. 5 Average recognition rate of $\pm 45^\circ$ pose of FERET database

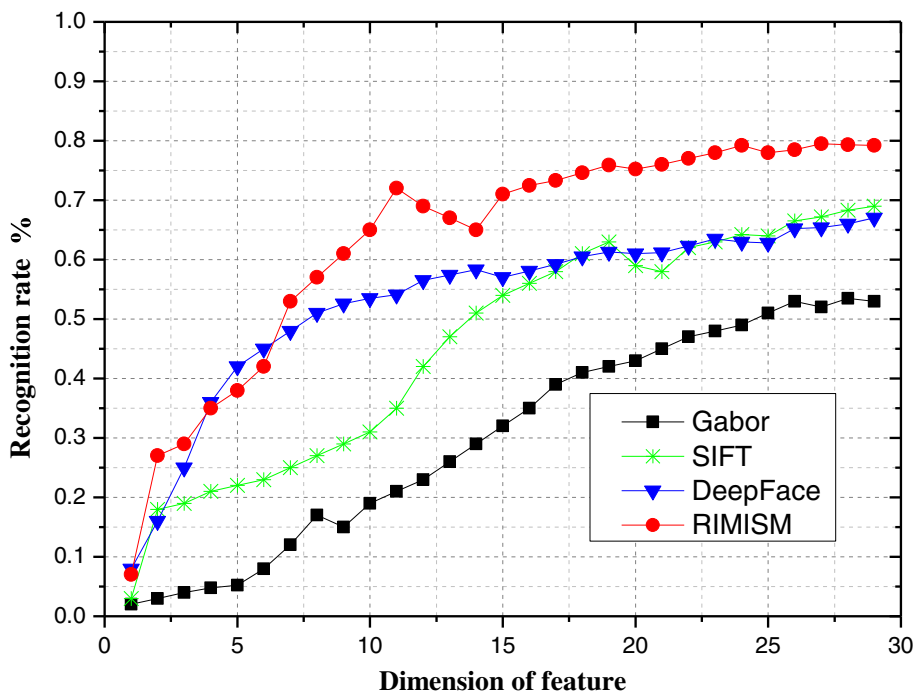


Fig. 6 Average recognition rate of $\pm 90^\circ$ pose of FERET database



Fig. 7 Pose change face image of MIT-CBCL database

the other three methods. There were two main reasons for the high efficiency of the RIM-ISM method:

- (1) Through constant iterations, learning a series of decreasing directions and scales in that direction, the initial shape was regressed to the position which was close to or equal to that of the real shape, so as to make the objective function converge to the minimum value at a very fast speed. Therefore, it avoided the problem of solving the Jacobian matrix and Hessian matrix, improved the running speed of the algorithm, and simplified the complexity of the algorithm.
- (2) Through using the interactive subspace method, the eigenvalue problem of high-dimensional matrix was transformed into the eigenvalue of a low-dimensional matrix, which greatly reduced the computation and improved the recognition efficiency.

To sum up, the RIM-ISM algorithm had strong robustness and higher recognition efficiency when dealing with random cross-pose samples.

5 Conclusion

Aiming at the problem of pose change recognition in face recognition technology, this paper proposed a cross-pose face recognition method based on regression iterative method and interactive subspace method.

Table 1 10-fold cross-validation comparison of MIT-CBCL database

Method	Average recognition rate (%)	Average recognition time
Gabor [11]	92.13 (115 × 115)	85.71 ms
SIFT [9]	95.42 (115 × 115)	102.63 ms
DeepFace [1]	96.25 (115 × 115)	68.42 ms
RIM-ISM	97.52 (115 × 115)	36.28 ms

The numbers in brackets indicate the image size

Through the regression iteration, the target function converges rapidly, and 45 pairs of important characteristic positions such as eyes, mouth, and nose of the target face sample were extracted. Later, the pose correction was carried out by solving the pose deflection angle of the cross-pose face image. Finally, the similarity of cross-pose face image was measured by using the interactive subspace method. The cross-pose recognition experiments in FERET database and MIT-CBCL database indicated that the RIM-ISM method proposed in this paper had a higher recognition accuracy, and it simplified the computation complexity. Meanwhile, it had a strong robustness in dealing with face samples with large angle pose changes.

Abbreviations

ISM: Interactive subspace method; MVP: Multi-view perceptron; PAMs: Pose-aware CNN models; RIM: Regression iterative method; SIFT: Scale-invariant feature transform; SPAE: Stacked progressive auto-encoders

Acknowledgements

This research had received support from the Science and Technology Department of Henan Province.

Funding

The authors acknowledge the National Natural Science Foundation of China-Henan Joint Fund (Grant: U1804162), the Henan Province Science and Technology Project (Grant: 192102310180).

Availability of data and materials

Not applicable.

Authors' contributions

LKF is the main writer of this paper. He completed the derivation process of RIM-ISM cross-pose face recognition algorithm, completed the simulation in FERET database and MIT-CBCL database, and analyzed the experimental result. HQZ provided some important suggestions for face pose estimation and experimental methods. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 19 February 2019 Accepted: 10 April 2019

Published online: 02 May 2019

References

1. Y. Taigman, M. Yang, M. Ranzato, in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference*. DeepFace: Closing the gap to human-level performance in face verification (Columbus, 2014), pp. 1701–1708
2. I. Masi, S. Rawls, G. Medioni, in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference*. Pose-aware face recognition in the wild (Las Vegas, 2016), pp. 4838–4846
3. Z. Zhu, P. Luo, X. Wang, in *the 27th International Conference on Neural Information Processing Systems*. Multi-view perceptron: a deep model for learning face identity and view representations (Cambridge, 2014), pp. 217–225
4. M. Kan, S. Shan, H. Chang, in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference*. Stacked Progressive-Encoders (SPA) for face recognition across poses (Columbus, 2014), pp. 1883–1890
5. J. Yim, H. Jung, B.I. Yoo, in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference*. Rotating your face using multi-task deep neural network (Boston, 2015), pp. 676–684
6. Y. Sun, X. Wang, X. Tang, in *Computer vision and pattern recognition (CVPR), 2014 IEEE Conference*. Deep learning face representation from predicting 10,000 classes (2014), pp. 1891–1898
7. F. Schroff, D. Kalenichenko, J. Philbin, in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference*. FaceNet: A unified embedding for face recognition and clustering (Boston, 2015), pp. 815–823
8. O.M. Parkhi, A. Vedaldi, A. Zisserman, in *British Machine Vision Conference*. Deep face recognition, vol 171 (2015), pp. 1–171 12
9. L. Wu, Z. Peng, Y. Hou, et al., Complete pose binary SIFT for face recognition with pose variation. *Chin. J. Sci. Instrum.* **36**(4), 736–742 (2015)
10. C. Li, W. Wei, J. Li, et al., A cloud-based monitoring system via face recognition using Gabor and CS-LBP features. *J. Supercomput.* **73**(4), 1532–1546 (2017)
11. B.S. Oh, K.A. Toh, A. Teoh, et al., An analytic Gabor feedforward network for single-sample and pose-invariant face recognition. *IEEE Trans. Image Process.* **27**(6), 2791–2805 (2018)
12. R.S. Babatunde, S.O. Olabiyisi, E.O. Omidiora, Assessing the performance of random partitioning and K-fold cross validation methods of evaluation of a face recognition system. *Adv. Image Video Process.* **3**(6), 18–26 (2015)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
