A NEW ATTEMPT TO FACE RECOGNITION USING 3D EIGENFACES

Chenghua Xu[1], Yunhong Wang[1], Tieniu Tan[1], Long quan[2]
[1]Center for Biometric Authentication and Testing, National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, P. R. China, 100080
E-mails: {chxu, wangyh, tnt }@nlpr.ia.ac.cn
[2] Department of Computer Science, Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
E-mails: quan@cs.ust.hk

ABSTRACT

Face recognition is a very challenging issue and has attracted much attention over the past decades. This paper makes a new attempt to face recognition based on 3D point clouds by constructing 3D eigenfaces. First, a 3D mesh model is built to represent the face shape provided by the point cloud. Then, the principle component analysis (PCA) is used to construct the 3D eigenfaces, which describe each mesh model in a lower-dimensional space. Finally, the nearest neighbor classifier and K-nearest neighbor classifier are utilized for recognition. Experimental results on 3D_RMA, a likely largest 3D face database available currently, show that the proposed algorithm has promising performance with a low computational cost.

Keywords: 3D eigenfaces, mesh model, 3D point cloud, 3D face recognition

1. INTRODUCTION

Nowadays biometric identification has obtained a wide interest not only in the laboratory but also in civilian applications. Of all the biometrics features, face is among the most common and most reachable so that face recognition remains one of the most active research issues in pattern recognition. In the past decades, many works focus on the source of 2D intensity or color images. The recognition accuracy is sensitive to lighting conditions, expressions, viewing position and varieties of subordinates such as hair, glasses. So far, it is still difficult to develop a robust automatic 2D face recognition system.

The 3D facial data can provide more geometric information for recognition than that of 2D images and has potential possibility to improve the performance of the system. With the development of 3D acquisition system, 3D capture is becoming faster and cheaper, and face recognition based on 3D information is attracting much attention. Some researches on curvature analysis [1,2,10] have been proposed for face recognition based on the high-quality range data from 3D laser scanners. In [3,4], a 3D morphable model was described with a linear combination of the shape and texture of multiple exemplars. This model could be fitted to a single image to obtain the individual parameters, which were used to characterize the personal features. Their results seemed very promising except that the modeling process incurred a high computational cost. Chen *et al.* [5] treated face recognition as a 3D non-rigid surface matching problem and divided the human face into rigid and non-rigid regions. The rigid parts are represented by point signatures to identify the individual. Beumier *et al.* [6,7] developed a 3D acquisition prototype based on structured light and built a 3D face database. They also proposed two methods of surface matching and central/lateral profiles to compare two instances. Both of them constructed some central and lateral profiles to represent the individual, and obtained the matching value by minimizing the distance of the profiles. It should be noted that there are two main difficulties facing 3D face recognition: high computational and spatial cost and inconvenient 3D capture. The existing methods usually have the high computational cost [3,4,6,7] or are tested on a small database [1,2,5,10].

Eigenfaces based on 2D images have been proposed by Turk *et al.* [8] and applied to face recognition successfully. 3D Eigenfaces are also introduced for face modeling by Iwasa *et al.* [9]. Here, we use 3D eigenfaces for face recognition based on 3D point clouds. First, 3D mesh models (called mesh image) are built to describe the geometric features of individual faces. Each 3D mesh model can be considered as a 2D image that the grey value is the depth value of the mesh nodes. Then, PCA method is used to obtain dominant eigen vectors, called 3D eigenfaces. Any new mesh model can be represented with the linear combination of these eigenfaces. Thus, one mesh model can be projected into the lower-dimensional space by these eigenfaces.

The main contributions of this paper are as follows: 1) A robust method is developed to build the mesh model based on the scattered point cloud. 2) 3D eigenfaces are constructed to realize face recognition on the 3D_RMA database [6]. Since matching process is performed in a lower-dimensional space, it has a low computational and spatial cost.

The remainder of this paper is organized as follows. In Section 2, we introduce how to obtain the mesh model from the 3D point cloud. The process of calculating 3D

eigenfaces is described in Section 3. Section 4 illustrates the classifiers for face recognition. Section 5 reports the experimental results and gives some comparisons with existing methods. Finally, Section 6 summarizes this paper and future work.

2. FACE MODELING

Each instance in 3D_RMA is represented with one 3D scattered point cloud. We intend to build a regular mesh with a fixed number of nodes and facets to represent the shape of one human face. Moreover, the different meshes have the corresponding nodes and same poses. Our modeling process includes three steps: pre-modeling, calculating transformation and modeling as outlined in Fig.1. In the following, we'll introduce them in detail.

2.1. Pre-modeling

Beginning with a simple basic mesh (see Fig.3a), a regular and dense mesh model is generated to fit the 3D scattered point cloud. We develop a universal fitting algorithm for regulating the hierarchical meshes to be conformed to the 3D points. This process includes two steps: initialization and hierarchical mesh fitting.

(1) Initialization of basic mesh

Our basic idea for modeling is to use optimization to tune the position of mesh nodes so that they are close to the point clouds perfectly. To keep the fitted result reasonable, one of the most pivotal problems is to give this fitting process a good initial value. Thus, it is necessary to detect some features correctly in the 3D scattered point clouds.

Some algorithms [2,10] have been proposed to label some facial features based on the high-quality data obtained from the laser scanner. They do not work on 3D_RMA data due to the limited quality, which has been proved in the literature [6,7]. Beumier *et al.* [6] also observed that the nose seems to be the only facial feature providing robust geometrical features for limited effort. We localize the prominent nose in the point cloud and utilize it to initialize the basic mesh as shown in Fig. 2a.

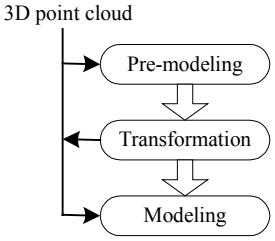


Figure 1. Modeling process

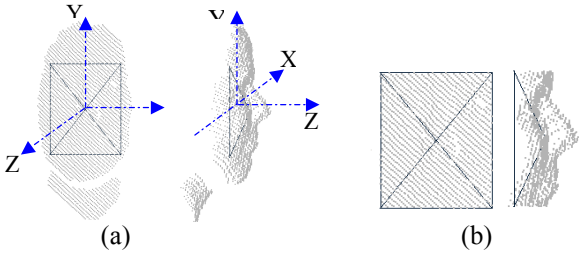


Figure 2. Initialization of the basic mesh. (a) The basic mesh is moved to the position of the nose tip in X-Y plane. (b) The points outside the basic mesh are ignored.

In most cases, there is much noise around the brim of the point cloud. To avoid the effect of the noise, we ignore the points whose projection on X-Y plane is out of the basic mesh as shown in Fig.2b.

(2) Hierarchical mesh fitting

After initialization, the basic mesh is aligned with the point cloud. Nevertheless, the basic mesh is so coarse that the basic contour of human face cannot be described. The subdivision scheme [11] is utilized to refine the basic mesh, and at the same time the refined mesh is regulated according to the data at each level. With the proceeding of refinement and regulation, the mesh can represent the individual well level by level.

Here, we describe the regulation in one refining level, which can be extended to all the levels. During the process of regulation, not only the nodes move forward to the 3D point cloud, but also the whole surface needs to be kept as smooth as possible. To meet with these two requirements, we define the following energy function:

$$E(\alpha) = E_{dis}(x_i, \alpha) + \omega E_{smooth}(\alpha) \qquad (1)$$

where x_i is a 3D point, α is a parameter vector and ω is the positive weighted factor. The distance term E_{dis} means the sum of weighted squared distances from the 3D data points to the model. The smooth term E_{smooth} attempts to keep the local area planar as described in Marschner *et al.* [12].

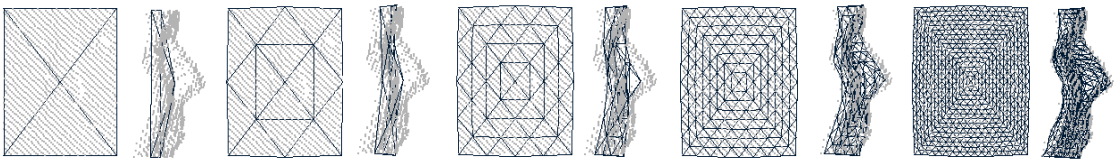
Minimizing $E(a)$ through regulating the fitted parameters a is a global optimization problem, which can be solved well by *Levenberg-Marquardt* method [13]. To reduce the dimension of the fitted parameters, we only regulate the Z position of each node instead of regulating all the coordinates.

Fig.3 shows the mesh after regulation in different refining levels. The mesh of level four is dense enough to represent the face surface. Of course, the denser the mesh is, the better the face is represented. Obviously, the denser mesh costs more time and space. In this paper, we use the mesh refined four times.

2.2. Obtaining transformation

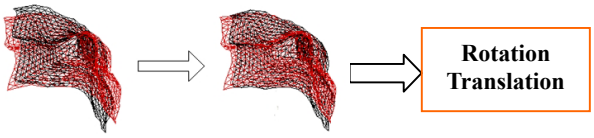
The different point clouds have different position and rotation relative to the 3D equipment. The mesh model obtained from the previous step has the same pose to its corresponding point cloud. Thus we can get the transformation parameters from the mesh models rather than from the point clouds directly, which will save much time.

First, an average mesh model is obtained by averaging



(a) (b) (c) (d) (e)

Figure 3. The regulated mesh models in different levels. (a) Basic mesh. (b) Level one. (c) Level two. (d) Level three. (e) Level four.



(a) (b) (c)

Figure 4. Transformation between the average model (black) and one individual model (red). (a) The average model and the individual model. (b) Models after transformation. (c) Transformation parameters.

the mesh models from pre-modeling process. This average model is considered as the ground model and all the models are rotated and translated to align with it.

We define an energy function

$$D(R,T)=\sum_{i=1}^{n} d_z(R,T;n_i) \qquad (2)$$

where R and T mean the rotation and translation parameters respectively, n_i is a node of the regulated model and d_z is the Euclidean distance from n_i to the average model.

There are six parameters (three translation and three rotation) to be considered. During the regulation, the rotation parameters are first tuned, and then the translation parameters (see Fig. 4 for this procedure).

Before this optimization process, we obtain a good initialization by moving both models so that their *nose tip nodes* (center node of the basic mesh) are on the origin of the coordinate system. We use *Levenberg-Marquardt* method [13] to solve this problem again. In this processing, the *nose tip node* gives a good initialization, which avoids the local minima effectively. This process can converge rapidly, about 0.5 seconds. Finally, we obtain the result that the models have the best superposition with the average model, as well as the values of rotation and translation.

2.3. Modeling

We transform the original point clouds using the obtained values from previous section so that they have proper alignment with the average mesh model. The transformed point cloud is modeled in the same way as the first modeling stage. Thus all these built mesh models have the same pose and represent the facial geometric shape realistically. Next we will use this kind of model to construct 3D eigenfaces.

Due to noise, wrong detection of nose tips and other unimagined reasons especially in automatic database in 3D_RMA, some built mesh models cannot describe the geometric shape of the individual. These mesh models are called non-face models. Fortunately, the non-face models are rare (<5% in automatic DB) and we can find these models using the reconstructed error described in the following section.

3. 3D EIGENFACES

Each point cloud can be represented with a regular 3D mesh containing many nodes (545 nodes in our case). Obviously, all the faces have similar general shape and will not be randomly distributed in this huge space. We can find a lower-dimensional space to exactly describe the facial distribution using the idea of eigenfaces [8].

We consider one 3D mesh as a two-dimensional intensity image. Each node of 3D mesh is regarded as a pixel of an intensity image and Z-coordinates of the mesh nodes are regarded as the intensity values. Thus a mesh image is generated, which is very similar to the 2D images. Each mesh model is represented with one d dimension vector M where d is the number of mesh nodes and each component is the value of Z-coordinates.

Let mesh images in the training set be represented by $M_1, M_2, \ldots, M_n$, where n is the number of the training meshes. The average 3D mesh model M_{aver} is calculated easily. Each mesh differs from the average with the vector $\Phi_i = M_i - M_{aver}$. Then one covariance matrix is constructed as follows

$$C=\frac{1}{n}\sum_{i=1}^{n}\Phi_i{\Phi_i}^T = AA^T \qquad (3)$$

The matrix C is 545 by 545 and we can obtain its eigenvalues and corresponding eigenvectors by solving an alternative matrix $A^T A$ (n by n) [8].

Generally, we can obtain $n-1$ non-zero eigenvalues and $n-1$ orthogonal eigenvectors. We can select the first $e(e<n)$ largest eigenvalues to approximate the facial geometric space and their corresponding eigenvectors are $u_1, u_2, \ldots, u_e$, which are called 3D

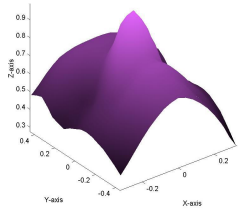


Figure 5. Average mesh image

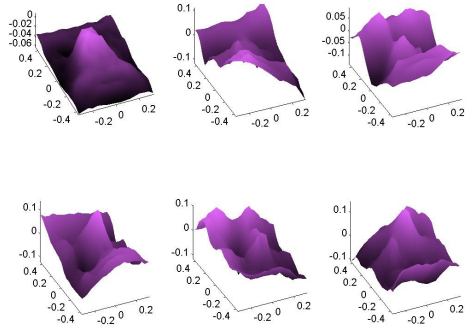


Figure 6. First six largest 3D eigenfaces

eigenfaces.

To show the results realistically, we use one database (Manual DB, session1, 90 instances) to obtain the 3D eigenfaces. Fig.5 shows the average mesh model, and Fig.6 shows the first six largest 3D eigenfaces. In our experiments, the first 20 eigenvalues occupy almost all the energy (98%).

To a new mesh model M, it can be transformed into this *e*-dimensional space by a simple operation

$$\omega_i = (M - M_{aver})^T u_i, i = 1,2,...,e. \quad (5)$$

Thus, each mesh model will become one point in the *e*-dimensional space.

We also reconstruct the mesh model according to the coefficients of the eigenfaces:

$$M_{recon} = M_{aver} + \sum_{i=1}^{e} \omega_i u_i \quad (6)$$

The reconstructed error can be described as

$$diff = (M - M_{recon})^T (M - M_{recon}). \quad (7)$$

With further observation, we can find that the reconstructed error is small if the mesh model is like the human face; otherwise it is large. Intuitively, we can preset a threshold, which determines whether one mesh is a face mesh or not. We test it on the set of the first 90 instances of automatic DB (session1) and the reconstructed error is showed in Fig.8. The meshes of No.14, 20, 37 and 38 have big reconstructed error and thus, we can determine that they do not describe faces well. The mesh model of No.38 and its reconstructed model are showed in the bottom row of Fig.7. During the recognition process, we can remove the non-face models

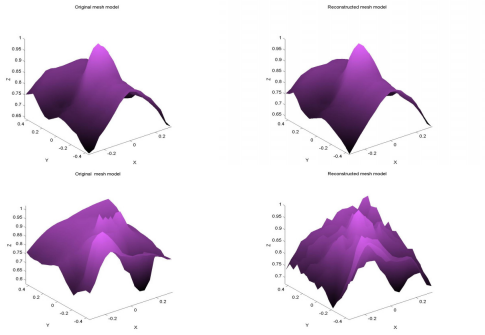


Figure 7. Original mesh models (left) and reconstructed models (right). The top row has small reconstructed error and the bottom has big error.

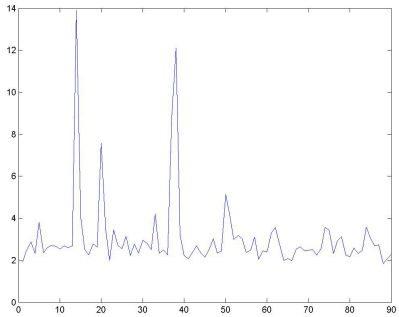


Figure 8. Reconstructed errors of the mesh models on automatic DB, seesion1 in 3D_RAM. Non-face models have big reconstructed error.

in order to avoid their influence.

4. FACE RECOGNITION

All the 3D point clouds in one database are approximated with 3D meshes, which are divided into the training and the testing sets. Then 20 most dominant 3D eigenfaces are obtained (98% of the total energy) from the training set. We transform the mesh models into this 3D eigenface space and each mesh model has one coefficient vector $V_i \in R^{20\times1}$, which forms the gallery set $(V_1, V_2, ..., V_n)$.

The testing mesh model is also projected into the same 3D eigenface space and forms the vector $V \in R^{20\times1}$, which is then compared to the vectors in the gallery set using the Euclidean metric $d_i = (V - V_i)(V - V_i)^T$. We use the nearest neighbor classifier (NN) and K-nearest neighbor classifier (KNN) respectively to determine which classification the tested sample belongs to. Here our focus is to validate the separability in the lower-dimensional space and only classifiers are used. More sophisticated classifiers may be applied to improve the recognition accuracy.

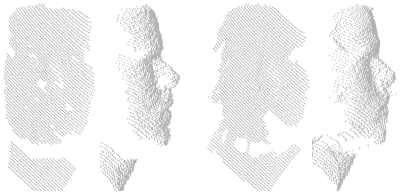
5. EXPERIMENTS

To demonstrate the performance of our proposed method, we implement it on the 3D database 3D_RMA. All these tests are finished under the hardware environment of PⅣ 1.3G CPU and 128M RAM.

5.1. 3D face databases

Our experiments are done on the 3D face database 3D_RMA [6, 7], where each face is described with a 3D scattered point cloud, obtained by structured light. Compared with the data obtained from the laser scanner, these point clouds are of limited quality (Fig.9).

The database includes 120 persons and two sessions: Nov. 97(session1) and Jan. 98 (session2). In each session, each person is sampled three shots, corresponding to central, limited left/right and up/down poses. People sometimes wear their spectacles, and beards and moustaches are also represented. From these sessions, two databases are built: automatic DB (120 persons) and manual DB (30 persons). Two 3D point clouds from manual DB and automatic DB are showed in Fig. 9 respectively and each shot shows the front and profile views. Here we can see that the quality of manual DB is better than that of automatic DB.



(a) (b)

Figure 9. 3D point cloud from manual (a) and automatic DB (b) of 3D_RMA respectively;

5.2. Experimental results

Identification accuracy is evaluated with the different sets in 3D_RMA. Considering the limited quantity of the samples, we use the method of Leave-one-out Cross Validation. In each data set, all the point clouds are represented with mesh models. Each time we leave one mesh image out as a test sample and train on the remainder. After computing the similarity differences between the test sample and the training data, the nearest neighbor (NN) and K-nearest neighbor (KNN) are then applied for classification. Table 1, Table 2 and Table 3 summarize the Correct Classification Rate (CCR) in manual DB, first 30 persons of automatic DB and all persons of automatic DB.

To avoid their influence of non-face models, we evaluate the CCR after removing them and the results are shown in the last two columns in Table 2 and Table 3.

In addition, we use one more familiar method, Cumulative Match Score (CMS) [14], to evaluate the identification performance. Fig.10 and Fig.11 show the CMS curves using the NN classifier on manual DB and automatic DB (first 30 persons and 120 persons, with and without non-face models) respectively.

Table 1. CCR in Manual DB (30 persons)

Database	NN (%)	KNN (%)
Manual DB, session1 (3 instances for each)	92.2	92.2
Manual DB, session2 (3 instances for each)	84.4	84.4
Manual DB, session1-2 (6 instances for each)	93.9	93.9

Table 2. CCR in Automatic DB (First 30 persons)

Database	First 30 models		Non-face meshes removed (22 persons)	
	NN (%)	KNN (%)	NN (%)	KNN (%)
Automatic DB, session1 (3 instances for each)	71.1	73.3	83.3	83.3
Automatic DB, session2 (3 instances for each)	80.0	80.0	89.4	89.4
Automatic DB (6 instances for each)	80.6	82.2	92.4	92.4

Table 3. CCR in Automatic DB (120 persons)

Database	All models (120 persons)		Non-face meshes removed (91 persons)	
	NN (%)	KNN (%)	NN (%)	KNN (%)
Automatic DB, session1 (3 instances for each)	59.2	60.3	71.1	71.8
Automatic DB, session2 (3 instances for each)	59.2	61.1	67.4	68.5
Automatic DB (6 instances for each)	69.4	71.1	79.3	80.2

From an overall view of Table 1-3, Fig.10-11, we can draw the following conclusions: a) Identification performance on manual DB is better than that on automatic DB since the data on manual DB has better quality (comparing Table 1 with Table 2). b) The non-face models in automatic DB affect the performance strongly, which is distinctly showed in Table 2-3 and Fig.11. This can remind us that the recognition performance can be improved by building better models further. c) The NN and KNN classifiers have similar performance in all the databases (see Table 1-3). d) The increase of the training samples can improve CCR (see Table 1-3 and Fig.10). e) The reconstructed error strongly affects the identification accuracy. The smaller the reconstructed error is, the higher the correct rate is. To explain this further, Fig.12 shows the CCR in different reconstructed error intervals on one automatic DB.

5.3. Comparisons

We make detailed comparisons with some existing methods to show the feasibility of our algorithm.

(1) Our method has a lower computational cost. Our modeling process costs more time (about 2s for each) while the matching process costs little time due to only calculating the Euclidean distance between two points in

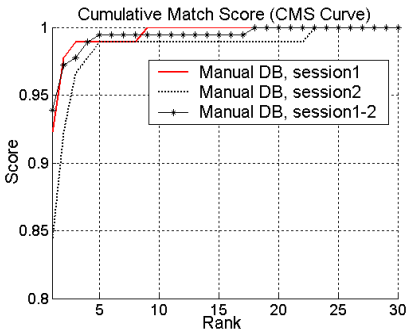


Figure 10. CMS curves on manual DB

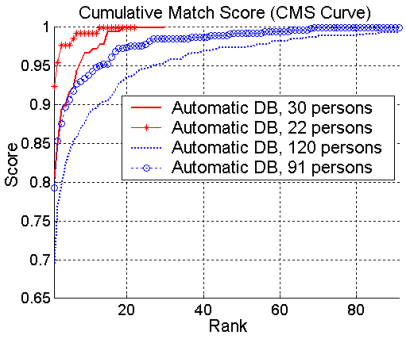


Figure 11. CMS curves on automatic DB

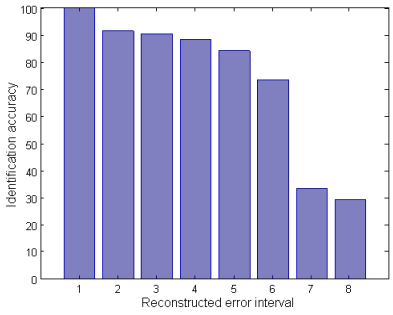


Figure 12. identification rate and reconstructed error.

Table 4. EER of our algorithm and automatic matching algorithms in Beumier *et al.* [6] on manual DB

Algorithms	Session1	Session2	Session1-2
Beumier [6], SURF	8.0%	7.0%	13.0%
Beumier [6], CLP	4.75%	6.75%	7.25%
Ours	7.0%	5.5%	5.0%

a lower dimensional space. Beumier *et al.* [6] develop surface matching (SURF) and central/lateral profiles (CLP) to realize face authentication and their verification performance is close to ours as shown in Table 4. However, their matching process is an optimization process, which incurs a high computational cost (at least 0.5s for each matching).

In [3,4], a 3D deformable generic model is developed to build the individual model according to a single image and the shape and texture parameters are applied to face recognition. Although their results seem promising, the modeling process is very slow: about 40 minutes.

(2) Our algorithm is tested on a bigger and more complex database. Gordon [2] obtained the higher recognition rate (100%) using depth and curvature features because they adopted a small database (only 8 persons) with high-quality range data (similar to Fig.10c) and is without eyeglasses, beards or pose difference.

Chua *et al.* [5] used the rigid region to characterize the individual in order to conquer the influence of the expressions. They tested their algorithm with only six objects (four expressions for each, without pose variations) and obtained promising results. Our algorithm is performed on 3D_RMA, which contains up to 120 persons with different pose and expression variations.

6. CONCLUSIONS

Inspired by eigenfaces based on 2D images, we propose 3D eigenfaces using statistical principal component analysis after constructing 3D mesh models to characterize the geometric features of the personal faces. The recognition is performed in the lower-dimensional space formed by 3D eigenfaces, which incurred a low computational and spatial cost. The proposed algorithm is tested with the 3D_RMA database, a likely largest 3D face database currently. The results and comparisons with previous works have demonstrated the feasibility of the algorithm. In the future, we will focus on improving the accuracy of mesh models as well as using other statistic methods to improve the identification rate.

Acknowledgements: This work is supported by research funds from the Natural Science Foundation of China (Grant No. 60121302 and 60332010) and the Outstanding Overseas Chinese Scholars Fund of CAS (No.2001-2-8). We would like to thank Dr. C. Beumier for the database 3D_RAM.

REFERENCES

[1] J.C. Lee, and E. Milios, "Matching Range Images of Human Faces", *Proc. ICCV'90*, pp.722-726, 1990.

[2] G.G. Gordon, "Face Recognition Based on Depth and Curvature Features", *Proc. CVPR'92*, pp.108-110, 1992.

[3] S. Romdhani, V. Blanz and T. Vetter, "Face Identification by Matching a 3D Morphable Model Using Linear Shape and Texture Error Functions", *Proc. ECCV'02*, pp.3-19, 2002.

[4] V. Blanz, S. Romdhani and T. Vetter, "Face Identification across Different Poses and Illumination with a 3D Morphable Model", *Proc. FG'02*, pp.202-207, 2002.

[5] C.S. Chua, F. Han, and Y.K. Ho, "3D Human Face Recognition Using Point Signiture", *Proc. FG'00*, pp.233-239, March, 2000.

[6] C.Beumier and M.Acheroy, "Automatic 3D Face Authentication", *IVC*, 18(4):315-321, 2000.

[7] C.Beumier and M.Acheroy,"Automatic Face Authentication from 3D Surface", *Proc. BMVC'98*, pp.449-458, 1998.

[8] M. Turk, and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, 3(1): 71-86, 1991.

[9] T. Iwasa, T. Shima, M. Sai, and G. Xu, "3D Eigenfaces for Face Modeling", *Proc. ACCV'02*, Jan. 2002.

[10] Y. Yacoob and L.S. Davis, "Labeling of Human Face Components from Range Data", *CVGIP: Image Understanding*, 60(2):168-178, 1994.

[11] D. Zorin and P. Schroder, "Subdivision for Modeling and Animation", *SIGGRAPH Course Notes*, 2000.

[12] S.R. Marschner, B. Guenter, and S.Raghupathy, "Modeling and Rendering for Realistic Facial Animation", *Proc. 11th Eurographics rendering workshop*, pp.231-242, 2000.

[13] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W. T. Vetterling. *Numerical recipes in C: the art of scientific computation*, Cambridge Univ. Press, Cambridge, 1992.

[14] P.J.Phillips, H.Moon, S.A.Rizvi, and P.J.Rauss, "The Feret Evaluation Methodology for Face-Recognition Algorithm", *IEEE Trans. on PAMI*, 22(10):1090-1104, 2000.