Available online at www.sciencedirect.com
ScienceDirect
NEUROCOMPUTING
Neurocomputing 70 (2006) 597–602
www.elsevier.com/locate/neucom

Letters

Three-dimensional surface registration: A neural network strategy

Heng Liu[a,b], Jingqi Yan[a,*], David Zhang[c]

[a]Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200030, PR China
[b]Southwest University of Science and Technology, Mianyang 621000, PR China
[c]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, PR China

Received 18 December 2005; received in revised form 3 April 2006; accepted 3 April 2006
Communicated by R.W. Newcomb
Available online 6 June 2006

Abstract

Three-dimensional surface registration is a necessary step and widely used in shape analysis, surface representation, and medical image-aided surgery. Traditional methods to fulfill such task are extremely computation complex and sometimes will obtain bad results if configured with unstructured mass data. In this paper, we propose a novel neural network strategy for efficient surface registration. Before surface registration, we use mesh PCA to normalize 3D model coordinate directions. The results and comparisons show that such neural network method is a promising approach for 3D surface registration.

Keywords: Neural network; Surface registration; Mesh PCA; ICP

1. Introduction

Neural networks are able to extract latent rules from large amounts of chaotic, noisy data, making them especially suitable for processing 3D point clouds and triangle meshes so as to obtain geometrical 3D characteristics. Other related applications of neural networks include using a neural network for surface reconstruction [1,2] and for stereo correspondence and motion estimation in image processing and analysis [3–5]. To date, however, there has been little work on the use of neural networks at an essential step in 3D object recognition and representation [6,7], 3D surface registration.

3D surface registration can be implemented via a number of methods [8–10], a typical one being ICP (iterative closest point) [8]. Given a set of source points P and a set of destination model points X, the goal of ICP is to find the rigid transformation T that best aligns P with X. Beginning with an initial guess of the registration T_0, this algorithm iteratively calculates a sequence of transformations T until the registration converges. Although ICP is a numerical algorithm, when dealing with massive unstructured noisy data, it can be computationally costly or, in surface matching procedures, it can even fail. Some improvements to ICP have been proposed [11,12] which are based on certain initial estimates or suitable searching organization; however, due to the difficulty of accurately estimating initial positions and choosing proper distances, improvements in the performance of ICP have been limited.

In this work, we construct a three layer neural network to infer the corresponding transformation matrix T to match source and destination model surfaces. Since a neural network has powerful supervised learning ability, we replace the choice of initial position estimation and iterative computing as ICP does with neural learning and weight updating to get a final optimal transformation matrix. The disposal flow of our method is, firstly, we use mesh PCA [13] to normalize different model coordination directions; then we construct a neural network to deal with 3D models to acquire fine registration results. Our whole neural network 3D surface registration procedure is shown in Fig. 1.

The remainder of this paper is organized as follows. In Section 2, we describe the process of 3D model preprocessing and mesh PCA. In Section 3, we describe

*Corresponding author. Tel.: +86 21 3420 4031; fax: +86 21 3420 4022.
E-mail addresses: hengliu@sjtu.edu.cn (H. Liu), jqyan@sjtu.edu.cn (J. Yan), csdzhang@comp.polyu.edu.hk (D. Zhang).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.04.004

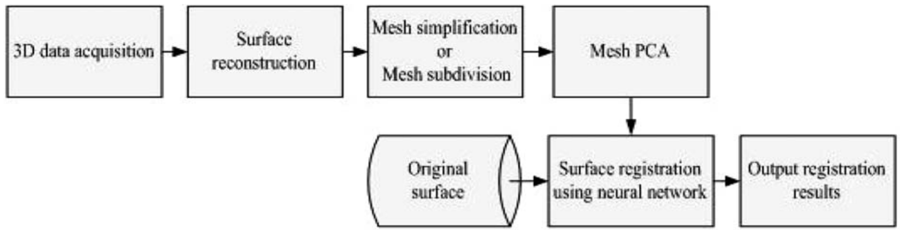


Fig. 1. A proposed 3D surface registration flow.

the neural network surface registration approach in detail. In Section 4 we provide our experimental results and compare our proposed approach with ICP in detail. Finally, in Section 5 we give a short conclusion.

2. 3D model preprocessing and mesh PCA

In this section, we first describe our process on 3D model preprocessing, which involves two main tasks, data acquisition and surface preprocessing. Then we give a brief description on mesh PCA.

2.1. 3D data acquisition and preprocessing

3D data can be obtained from 3D digitized devices or reconstruction and saved as 3D points in a VRML model file (.wrl file) rather than in a cloud file (.asc file). Objects in an .asc file consist only of point coordinate information in which surface information is only implicit. In contrast, in a .wrl file, objects are regarded as geometrical entities from which it is possible to directly extract the information of entire vertices and triangular faces. In this work, all of the experimental 3D data comes from multi-view reconstruction or range scans. Although most of the experiments and comparisons use reconstructed 3D materials, the methods described in this paper are also suitable for range scan 3D data.

Both the reconstruction and range scanning processes will introduce vertex noise, especially affecting complex 3D shapes. Topological noise can also be introduced if the surface is reconstructed from a 3D points cloud using Delaunay Triangulation principles [14]. To remove such kinds of noise we can use a surface denoise filter such as a Wiener filter [15]. After wiping off noise, we proceed to re-sampling different 3D models to make them keep the approximate model precision. There are two possible approaches aiming to this assignment. In one approach the low precision surface is subdivided [16] so that it aligns with the complex surface; in the other approach, the high precision surface is reduced [17] in order to achieve alignment with the low precision one.

Some 3D ear models acquired by reconstruction and preprocessing results are illustrated in Fig. 2.

2.2. Mesh PCA

Mesh PCA [13] uses Principal Component Analysis (PCA) to analyze vertex data and to extract the principal

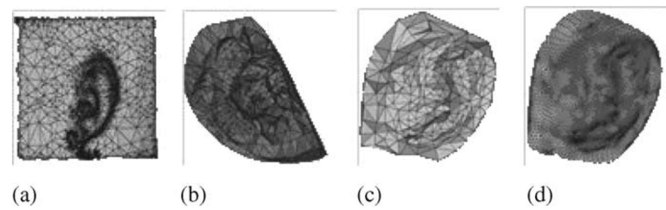


Fig. 2. 3D ear reconstructed modes and processing results. (a) Ear model containing vertex noise; (b) ear model containing topological noise; (c, d) ear models are denoised and correspondingly subdivided.

directions of the three largest spreads of the distributions of these vertices. A mesh PCA transformation allows the original coordinate frame axes to be changed to new axes while leaving the geometric shape of the 3D model unchanged. This means that adjusting the principal axes in three dominate directions allows us to coarsely match those 3D models represented by different coordinate frames in a unified coordinate frame. The covariance matrix C used as a PCA caused matrix is defined as

$$C = \frac{1}{\sum w_k} \sum_{k=1}^{N} w_k \left(\vec{v}_k - \vec{m}\right) \cdot \left(\vec{v}_k - \vec{m}\right)^{\mathrm{T}}, \tag{1}$$

where $\vec{v}_k$ is a vertex of a mesh and k is the vertex index, $\vec{m} = \frac{1}{N}\sum_{k=1}^{N} w_k \vec{v}_k$ is the centroid of the mesh, $w_k = (N \cdot S_k/3S)$ is the weight coefficient, S_k is the sum of surfaces of all triangles that have $\vec{v}_k$ as a vertex, S is the surface area of the mesh (i.e. the sum of the areas of all triangles in the mesh), and N is the number of vertices. After the K-L transformation [18] to get the three eigenvectors of C, we may normalize the eigenvectors and regard such three normalized eigenvectors as a 3×3 coordinate transformation matrix H. Thus we can utilize the matrix H to transform the original 3D model to a new form model in which coordinate directions have been changed but the geometric shape is not altered. That means we normalized the 3D model direction by a unified coordinate frame.

Fig. 3 shows two reconstructed 3D ear models and the corresponding adjusted results after the coordinate directions have been normalized using mesh PCA.

3. Neural network surface registration

In this section, we will give a detailed description on constructing a neural network for surface registration. First, we deduce mathematical principles for surface registration under the assumption of being a rigid

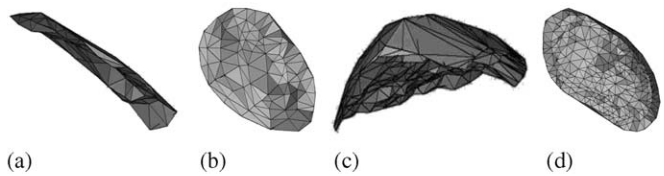


Fig. 3. Mesh PCA coordinate direction normalization: (a, c) original models; (b, d) mesh PCA results.

transformation. Then based on the principles, we construct a neural network structure for surface registration. At last, we discuss neural learning rules to update weights for the output converging.

3.1. Surface registration principles

Assuming 3D surfaces A and B have been adjusted by mesh PCA, we need to finish the A and B surface registration task. If A and B are both obtained from rigid objects, obviously, with the consideration of scale coefficients, such surface registration process can be regarded as a rigid transformation procedure. In our work, we treat surface registration as a complex parameter optimization procedure of rigid transformation. That means in order to match two surfaces A and B, we need to find one 3×3 rotation matrix R and one translation 3×1 vector T and one specific scale coefficient s to make a distance error function E to arrive at a minimum. This procedure can be defined as

$$\mathrm{Min}\, E = Dist(A, R(s\cdot B) + T), \tag{2}$$

where $\mathrm{Dist}(\bullet,\bullet)$ is a certain form distance function which can be defined according to concrete applications. Although Geodesic distance [10,19] is more suitable than Euclidean distance when measuring two corresponding vertices, we use Euclidean distance in neural network registration because it is easier to be represented with a neural network.

Then we assume that the summation of the distance error of every vertex pair in two surfaces represents the distance error between two such surfaces which need to be matched. Let $P_k(x,y,z)$ and $P'_k(x',y',z')$ be two vertices in two surfaces, $P'_{ek}(x',y',z')$ be a corresponding vertex against vertex $P'_k(x',y',z')$ according to the estimated transformation. Then we assume Eq. (2) to be expressed as

$$Min\, E = \sum_k \left\| P_k - P'_{ek} \right\|, \tag{3}$$

$$P'_{ek} = s\cdot RP'_k + T,$$

where k is the vertex index.

3.2. Neural network structure

To construct a suitable surface registration neural network, we will manage to find the corresponding relationship between the registration matrix and the weight matrix of the neural network.

Letting $R' = s\cdot R$, we can rewrite

$$P'_{ek} = R'P'_k + T \tag{4}$$

or

$$P'_{ek} = s\cdot RP'_k + T,$$

where W is the homogenous registration matrix

$$W = \begin{bmatrix} R' & T \\ 0 & 1 \end{bmatrix} \tag{5}$$

or equivalently

$$W = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

where $P'^{(H)}_k = [\,[P'_k \quad 1\,]^{\mathrm{T}} = \left[P'_{kx},\ P'_{ky}\ P'_{kz}\ 1\right]^{\mathrm{T}}$ is the corresponding homogeneous 3D point.

If we take the surface vertex coordinates as inputs, inspired by principles of 2D image matching [3–5], we can construct a three-layer neural network (shown in Fig. 4) based on the rule of least mean squared (LMS) error performance learning as in Eq. (3).

From Eq. (6), we get that

$$WP'^{(H)}_k = \begin{bmatrix} W_1 P'^{(H)}_k \\ W_2 P'^{(H)}_k \\ W_3 P'^{(H)}_k \\ 1 \end{bmatrix} = \begin{bmatrix} w_{11}P'_{kx} + w_{12}P'_{ky} + w_{13}P'_{kz} + w_{14} \\ w_{21}P'_{kx} + w_{22}P'_{ky} + w_{23}P'_{kz} + w_{24} \\ w_{31}P'_{kx} + w_{32}P'_{ky} + w_{33}P'_{kz} + w_{34} \\ 1 \end{bmatrix}. \tag{7}$$

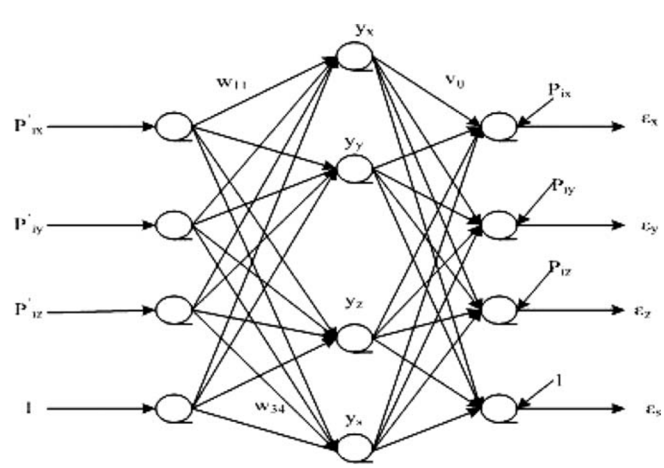


Fig. 4. A three-layer neural network to determine the registration matrix.

According to Eqs. (7) and (4), Eq. (3) can be rewritten to

$$Min\,E = \sum_k \left\| P_k - W \cdot P'_k \right\|$$
$$= \sum_k \sqrt{\left(P_{kx} - W_1 P'^{(H)}_k\right)^2 + \left(P_{ky} - W_2 P'^{(H)}_k\right)^2 + \left(P_{kz} - W_3 P'^{(H)}_k\right)^2}, \tag{8}$$

where k is the vertex index.

On the other hand, if x_{ij} represents the sense and strength of the connection between neuron j in the first layer and neuron i in the second layer, a 4×4 weight matrix X can be constructed between the first layer and the second layer, where

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Let $y_i = \begin{bmatrix} y_{ix} & y_{iy} & y_{iz} & y_{is} \end{bmatrix}^{\mathrm{T}}$ denote the actual outputs of layer 2. Then we can obtain

$$y_i = \begin{bmatrix} x_{11}P'_{ix} + x_{12}P'_{iy} + x_{13}P'_{iz} + x_{14} \\ x_{21}P'_{ix} + x_{22}P'_{iy} + x_{23}P'_{iz} + x_{24} \\ x_{31}P'_{ix} + x_{32}P'_{iy} + x_{33}P'_{iz} + x_{34} \\ 1 \end{bmatrix}. \tag{9}$$

Comparing Eq. (7) with Eq. (9) (subscript k and i have the same meaning: vertex indices), and noticing that we expect the output of layer 2 to be just the destination model vertex $P'^{(H)}_{ek}$, then we can deduce that the relationship between the weight matrix X and the unknown registration parameter matrix W is

$$W = X. \tag{10}$$

Since the weight matrix X can be computed through the performance learning process of a three-layer neural network as depicted in Fig. 4, when the neural network converges, the optimal registration matrix W can be acquired ($W = X$). Furthermore, based on the matrix W, we can take the following formulas to extract the rotation matrix R', the translation vector T:

$$R' = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}, \quad T = \begin{bmatrix} w_{14} & w_{24} & w_{34} \end{bmatrix}^{\mathrm{T}},$$

where

$$w_{11}^2 + w_{12}^2 + w_{13}^2 = 1,$$
$$w_{21}^2 + w_{22}^2 + w_{23}^2 = 1,$$
$$w_{31}^2 + w_{32}^2 + w_{33}^2 = 1$$

and

$$w_{11}w_{21} + w_{12}w_{22} + w_{13}w_{23} = 0,$$
$$w_{21}w_{31} + w_{22}w_{32} + w_{23}w_{33} = 0,$$
$$w_{31}w_{11} + w_{32}w_{12} + w_{33}w_{13} = 0.$$

If $w_{ij}(i,\ j = 1,\ 2,\ 3)$ in matrix W cannot satisfy the above constraints, we can use a typical matrix operation: perpendicular transformation to adjust the matrix W. After acquiring the rotation matrix R' from the W, according to the above formulas, we can naturally get the translation vector T. In addition, according to the distances ratio between two pairs of related vertices coming from the destination model and source model, respectively, and with the formula $R' = s \cdot R$, we can obtain the scale coefficient s and original rotation matrix R easily (more detailed information on estimating the R, T, s from W can be seen in [20]).

3.2. Neural learning

Neural learning refers to making the neural network weight matrix to update continually so that the output of the neural network can converge. In our surface registration procedure, we adopt the general gradient descending principles to guarantee update of the weight matrix.

Assuming the weight matrix V between the second layer and the third layer is defined as

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}, \tag{11}$$

where $V_i = [v_{i1}, v_{i2}, v_{i3}, v_{i4}], i \in (1,2,3,4)$ and v_{ij} denotes the sense and strength of the connection between neuron j in the second layer and neuron i in the third layer (see Fig. 4). Let $P_i = \begin{bmatrix} P_{ix} & P_{iy} & P_{iz} & P_{is} \end{bmatrix}^{\mathrm{T}}$ denote the ideal outputs of the second layer which are destination model vertices and $P'_i = \begin{bmatrix} P'_{ix} & P'_{iy} & P'_{iz} & P'_{is} \end{bmatrix}^{\mathrm{T}}$ denote the inputs of the neural network which are the source model vertices. According to Eq. (8), we should set the weight matrix V to be a negative unit matrix $v_{ij} = \begin{cases} -1 \text{if} i = j \\ 0 \text{ otherwise} \end{cases}$, in order to compare the destination vertex P_i with the affected source vertex P'_i by the first layer weight matrix W. And the output error ε of the neural network is obtained as

$$\varepsilon_i = \left[\varepsilon_{ix}, \varepsilon_{iy}, \varepsilon_{iz}, \varepsilon_{is}\right]^{\mathrm{T}}$$
$$= \left[P_{ix} - W_1 \cdot P'_i, P_{iy} - W_2 \cdot P'_i, P_{iz} - W_3 \cdot P'_i, 0\right]^{\mathrm{T}}. \tag{12}$$

Applying the LMS algorithm, the estimated gradients λ can be obtained as

$$\lambda_{ij} = \frac{\partial \varepsilon_{ij}^2}{\partial W_i} = -2\varepsilon_{ij} P'_i. \tag{13}$$

If we take a linear function as activation function of a neural unit, and use the gradient descending principle to update the weights between the second layer and the first layer: $W_{\text{new}} = W_{\text{old}} - \mu\lambda_{ij}$. Then the weight vector

$W_i, i \in (1,2,3)$ can be acquired

$$W_i^{n+1} = W_i^n + 2\mu\varepsilon_{ij}^n P_i', \tag{14}$$

where μ is the learning rate and n is the number of iterations. As for the homogenous weight coefficient W_4^n, it keeps the constant: $W_4^n = W_4^{n+1} = [0,0,0,1]$ in the whole neural learning procedure. The weight matrix V between the second layer and the third layer do not need to change in the learning process.

In practice, because the surface principal axes have been found by mesh PCA before applying the neural network, neural network training converges quickly in most cases. Once the neural network converges, the registration parameters matrix W can be obtained directly from the weight matrix X in terms of Eq. (10).

4. Experiments and discussions

In this section, we first conclude the entire surface registration transformation using a formula. Then we present our experimental results and make a detailed comparison and performance analysis to the ICP method. At last, we discuss some practical factors which affected the final registration results.

As we have stated above, to normalize the coordinate direction, 3D surface data must first be preprocessed using mesh PCA in order to acquire the initial coordinate direction transformation matrix H. Then we apply neural networks to find the registration matrix W.

In general, a vertex P_i' of the source surface will be required to take the full transformation $W \cdot H_i$ to get to the corresponding vertex P_i for the registration task. Such a transformation can be formulated as $P_i = (W \cdot H) \cdot P_i'$. Since 3D models have large faces as well as vertices, after the vertices have been transformed, we should adjust these faces directions using the same transformation matrix to guarantee consistent face visibility.

In our experiments, we test our methods both using reconstructed and range scan 3D data. Configured with the same 3D data, ICP is also used for the surface registration task. In the neural learning procedure, we set the learning rate μ to 0.05 and the largest iteration number to 3000. And in the ICP procedure, we choose the initial matching pairs randomly.

The results and the comparisons are shown in Figs. 5 and 6.

Fig. 5. 3D reconstructed ear surface registration. (a) The original high-precision reconstructed surface; (b) low-precision reconstructed surface; (c) surface registration using neural network strategy, time taken $t = 10$ s; (d) surface registration using ICP, time taken $t = 40$ s.

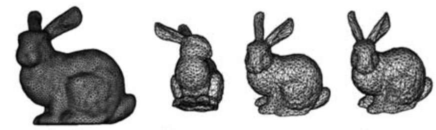


Fig. 6. Range data bunny surface registration. (a) The original high-precision range model; (b) corresponding mesh simplification model; (c) surface registration using neural network, time taken $t = 27$ s; (d) surface registration using ICP, time taken $t = 121$ s.

From Figs. 5 and 6, we can see that the models in Fig. 5(c) and Fig. 6(c) hold a more accurate pose compared to the destination models in Fig. 5(a) and Fig. 6(a) than there in Fig. 5(d) and Fig. 6(d). Accompanied with this, the time cost in neural network registration is less than that in ICP.

For all the test data available, the network surface registration method actually converges more quickly than the ICP procedure does. According to our counting, the average iteration number of the neural network strategy is always under 300 and the computation time is always within 40 s, whereas, ICP always takes more than 40 s, and sometimes, ICP fails. Moreover, using the same LMS rule, the error rate for neural network registration is under 3%, whereas for ICP the error rate can rise to 6%.

In addition, the ICP method cannot process the scale problem in a surface registration procedure while our neural network strategy can deal with this issue by easily setting scale coefficients. Further comparison with improved ICP methods need more materials to be supplied. However, because our method utilizes the good performance for supervised learning from a neural network, it can be expected to get good registration results in the further comparisons.

Furthermore, in a practical surface registration process, we find that mesh PCA also plays an important role on the final registration results. In addition, our surface registration strategy is based on the condition that two models have been pre-processed to contain similar sampling precision and vertex noise and topical noise have been basically wiped out. If 3D models are not preprocessed well, final surface registration results will be uncertainty. Even, sometimes, the neural network cannot converge at all.

5. Conclusions

In this paper, we propose an entire novel neural network strategy to deal with the 3D surface registration task. Experimental results and comparisons show our approach is more efficient than the traditional ICP method. Further work will focus on seeking a robust way to reduce model precision dependency and to use the geodesic distance representation for the neural network to measure vertex distance of different kinds of manifolds.

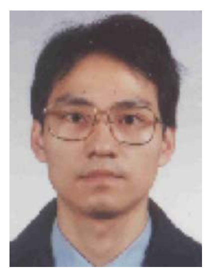
Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 60402020). We would like to thank Computer Vision Research Laboratory of the University of Notre Dame and the Stanford Computer Graphics Laboratory for providing their range scan data. In addition, we would like to thank Robert Newcomb and Martin Kyle for their good suggestion and kind help in writing this paper.

References

[1] A.A. Adan, M.A. Flexible, Similarity measure for 3D shapes recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1507–1520.
[2] J. Barhark, A. Fishcher, Adaptive reconstruction of freeform objects with 3D SOM neural networks grids, in: Proceedings in Pacific Graphics, Tokyo, 16–18 October 2001, IEEE Computer Society Press, silver Spring, MD, 2001,pp. 97–105.
[3] P. Besl, N. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.
[4] T. Chen, W.-C. Lin, C.-T. Chen, Artificial neural networks for 3-D motion analysis—Part I: Rigid motion, IEEE Trans. Neural Networks 6 (6) (1995) 1386–1393.
[5] J. Feldmar, G. Malandain, J. Declerck, N. Ayache, Extension of the ICP algorithm to nonrigid intensity-based registration of 3D volumes, Comput. Vision Image Understanding 66 (2) (1997) 193–206.
[6] D.A. Forsyth, J. Ponce, Compute Vision: a Modern Approach, Prentice-Hall, Englewood Cliffs, NJ, 2003.
[7] M. Garland, Y. Zhou, Quadric-based simplification in any dimension, Trans. Graphics 24 (2) (2005) 271–292.
[8] P.L. George, Improvements on Delaunay-based three-dimensional automatic mesh generator, Finite Elements Anal. Des. 25 (3–4) (1997) 297–317.
[9] F. Mokhtarian, M. Bober, Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization, Kluwer Academic Publishers, Dordreht, 2003.
[10] M.N. Nasrabadi, Y.C. Choo, Hopfield network for stereo vision correspondence, IEEE Trans. Neural Networks 3 (1) (1992) 5–13.
[11] J. Peng, V. Strela, D. Zorin, A simple algorithm for surface denoising, in: Proceedings of IEEE Visualization, San Diego, CA, USA, 2001, pp. 107–112.
[12] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: Proceedings of Third International Conference on 3D Digital Imaging and Modeling, 28 May–1 June 2001, Quebec City, Canada, 2001, pp. 145–152.
[13] M.S. Yamany, A.A. Farag, Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1105–1120.
[14] P. Schroder, Subdivision as a fundamental building block of digital geometry processing algorithms, J. Comput. App. Math. 149 (1) (2002) 207–219.
[15] Y. Sun, J. Paik, A. Koschan, D.L. Page, M.A. Abidi, Point fingerprint: a new 3-D object representation scheme, IEEE Trans. System Man Cybernet. Part B: Cybernetics 33 (4) (2003) 712–717.
[16] D. Tzovaras, N. Ploskas, M.G. Strintzis, Rigid 3-D motion estimation using neural networks and initially estimated 2-D motion data, IEEE Trans. Circuits Systems Video Technol. 10 (1) (2000) 158–165.
[17] D.V. Vranic, D. Saupe, J. Richter, Tools for 3D object retrieval: Karhunen–Loeve transform and spherical harmonics, in: Proceedings of IEEE 2001 Workshop Multimedia Signal Processing, October 2001, Cannes, France, 2001, pp. 293–298.
[18] A.R. Webb, Statistics Pattern Recognition, second ed., Wiley, New York, 2002.
[19] L. Wen Peng, Siti Mariyam Shamsuddin, 3D object reconstruction and representation using neural networks, in: Proceedings GRAPHITE 2004, 15–18 June, Singapore, 2004, ACM Press, New York, 2004, pp. 139–147
[20] D. Zhang, Harmonic shape images: a 3-D free-form surface representation and its application in surface matching. Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA, 1999.

Heng Liu, graduated from Department of Electric Automation in Southwest University of Science and Technology in 1997, and he joined the same university as a lecture from 1998 to 2001. And in 2004, he received his M.Sc. degree from Chongqing University. Now he is pursuing Ph.D. degree in the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. His current research interests include computer vision, pattern recognition and machine learning.



Jingqi Yan received his B.S. degree (1996) in automatic control, his M.S. (1999) and Ph.D. (2002) degrees in pattern recognition and intelligent system from Shanghai Jiao Tong University, Shanghai, China. He is presently a researcher in the Institute of Image Processing and Pattern Recognition at Shanghai Jiao Tong University. His interests include geometric modeling, computer graphics, biometrics, scientific visualization and image processing.

David Zhang graduated in Computer Science from Peking University in 1974. He received his M.Sc. and Ph.D. in Computer Science from the Harbin Institute of Technology (HIT) in 1982 and 1985, respectively. From 1986 to 1988 he was a Postdoctoral Fellow at Tsinghua University and then an Associate Professor at the Academia Sinica, Beijing. In 1994 he received his second Ph.D. in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada. Currently, he is a Chair Professor, the Hong Kong Polytechnic University where he is the Founding Director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR Government. He also serves as Adjunct Professor in Tsinghua University, Shanghai Jiao Tong University, Beihang University, Harbin Institute of Technology, and the University of Waterloo. He is the Founder and Editor-in-Chief, International Journal of Image and Graphics (IJIG); Book Editor, Kluwer International Series on Biometrics (KISB); and Program Chair, the International Conference on Biometrics Authentication (ICBA), Associate Editor of more than ten international journals including IEEE Trans on SMC-A/SMC-C, Pattern Recognition, and is the author of more thanbooks. Professor Zhang is a current Croucher Senior Research Fellow and Distinguished SpeakerComputer Society.