

Research Article

A Dense Point-to-Point Alignment Method for Realistic 3D Face Morphing and Animation

Yongli Hu, Mingquan Zhou, and Zhongke Wu

College of Information Science and Technology, Beijing Normal University, Beijing 100875, China

Correspondence should be addressed to Yongli Hu, hu_yongli00@sina.com

Received 29 January 2009; Accepted 13 March 2009

Recommended by Suiping Zhou

We present a new point matching method to overcome the dense point-to-point alignment of scanned 3D faces. Instead of using the rigid spatial transformation in the traditional iterative closest point (ICP) algorithm, we adopt the thin plate spline (TPS) transformation to model the deformation of different 3D faces. Because TPS is a non-rigid transformation with good smooth property, it is suitable for formulating the complex variety of human facial morphology. A closest point searching algorithm is proposed to keep one-to-one mapping, and to get good efficiency the point matching method is accelerated by a KD-tree method. Having constructed the dense point-to-point correspondence of 3D faces, we create 3D face morphing and animation by key-frames interpolation and obtain realistic results. Comparing with ICP algorithm and the optical flow method, the presented point matching method can achieve good matching accuracy and stability. The experiment results have shown that our method is efficient for dense point objects registration.

Copyright © 2009 Yongli Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Constructing alignment of 3D objects is a crucial element of data representations in computer vision and graphics. Generally the dense alignment is a point-to-point mapping from one surface onto another surface, where each point gets the correspondent point according to its inherent property, such as the points of nose tip on different 3D faces are correspondent points according to the feature of human face. However, the practices and applications of dense point correspondence have been increasing over the last years. The straightforward application of the dense alignment is to compute objects morphing and animation. More important, if the point correspondence of a class of objects has been established, it is achievable to construct a representation for these objects. The most typical and simple model is the linear combination model described in [1], where a 3D face morphable model was constructed on the aligned 3D faces, and given a facial image the 3D face can be reconstructed by a model matching procedure. The other applications, involving objects recognition based on 2D/3D images, shape retrieval, and 3D surface reconstruction in computer vision, are all relied on dense surface correspondence.

For dense 3D objects, as the complexity of model structure and the hugeness of data, it is a challenging problem to get good correspondence result, especially to high-resolution scanned 3D faces. In fact, the correspondence of different 3D faces is not a well-defined problem. When two faces are compared, only some distinct feature points, such as the tip of nose, the corner of mouth, and the center of eyes, have the clearly correspondent points, while it is difficult to define the correspondence for the points on the smooth regions, such as the cheeks and the forehead. However, even matching the distinct feature points may be a difficult problem because it involves many of the basic problems of computer vision and feature detection. To conquer the correspondence problem of dense 3D faces, we present a closest point matching method based on the thin plate spline (TPS) transformation. In this method, the source 3D face is firstly transformed onto the destination 3D face by TPS transformation, which is constructed from the interpolation on the feature points hand-placed on the source and target 3D face. Then using a revised closest point matching algorithm, the point-to-point alignment between 3D faces is obtained. We create 3D face morphing and animation from the interpolation between

the aligned 3D faces. The realistic deformation results and the experiments comparing with the related methods show that our correspondence algorithm may be an appropriate approach.

The remainder of the paper is structured as follows. In Section 2 we review some related work. In Section 3 the TPS transformation of 3D faces is described in detail. Then the point-to-point alignment is established in Section 4. In Section 5, 3D face morphing and animation are implemented, and experimental results are given. Finally this work is concluded.

2. Related Work

In the past decades, there are many methods and algorithms that are presented to solve surface alignment and dense point correspondence for different applications. All these researches fasten on two element problems about the point matching: the spatial transformation and feature correspondence searching. The former one is to find a suitable transformation for the aligning objects. These spatial transformations can be classified into rigid transformation and nonrigid transformation. The rigid transformation is generally used in the alignment of an object and itself, such as the different viewpoint scenes or the overlapped parts of the object. The nonrigid transformation, including affine transformation, spline function, and radial-based function, now is the dominant method used in the cases existing nonrigid deformation. The latter issue of point alignment generally concerns how to determine the right correspondence by the inherent features of the objects, which commonly have the forms in geometry properties, like points, lines, curves, and surfaces, or the abstract measurements, such as moment, entropy, and mutual information. There are several surveys [2–6] that have given comprehensive reviews about this subject. The following are some typical work related to our method.

One of the most popular point matching methods is the iterative closest point (ICP) algorithm proposed by Besl and McKay [7]. It iteratively searches for closest points in two surface patches and optimizes the rigid transformation to minimize the average distance of these closest points. The original ICP algorithm demands adequate prealignment and does not usually guarantee the one-to-one correspondence, as a result various improved ICP methods were proposed. Rusinkiewicz and Levoy provided good surveys over these ICP variants [8]. Although these improvements have enhanced the convergence of ICP and achieved high registration accuracy, the rigid transformation constrains its application. In many nonrigid deformation cases, ICP is not suitable, such as 3D faces.

Blanz and Vetter made dense correspondence between 3D facial scans [1, 9], taking advantage of the fact that the radial coordinate from Cyberware scans can be expressed as a height map image with the intensity representing the radius in cylinder coordinate system. They used optical flow technique to establish correspondence between texture images and height maps images, and the correspondence

was refined by a bootstrapping method if large amount of the prototypic scans obtained. A 3D face representation named morphable model was constructed from the set of aligned 3D faces. Recently, they proposed a new dense 3D correspondence method [10] based on their 3D faces database. In this method, a facial feature learning strategy and automatic properties extraction algorithm were used for alignment optimization. Although their alignment has convincing results, it demands large quantities of 3D facial scans, and some 3D information will be lost when the alignment is perceived from 2D images optical flow computation.

Similarly, the notable TPS-RPM method of Chui and Rangarajan [11] attempted to incorporate TPS into the framework of ICP for point matching. A binary correspondence matrix was used in this method to record the matching relation of all points and eliminate outliers. In point matching procedure, a soft-assign and deterministic annealing optimization was implemented to compute point correspondence iteratively. Although their experiments show good results on some sparse 2D/3D point sets, the method can easily get trapped in bad local minima if the objects are not approximately aligned initially [12]. And this method is not suitable for the alignment of 3D faces with large quantity of dense points because of the limitation of the dimension of the correspondence matrix and the impracticalness of applying TPS on the whole dense point sets.

The interpolation idea in [13] is very close to our method. To synthesis facial expression from photographs, a general 3D facial model was fitted to the individual faces based on radial basis functions using 13 feature points [13]. But the general 3D facial model created by Alias—Wavefront tools—is a relative sparse model comparing with the dense 3D faces. In addition, the fitting procedure and its refinement are different from the closest point matching algorithm here.

There are other researches associated with surface or dense point correspondence, but the applications are various. The medical image registration may be the dominant domain, others applications include 3D objects reconstruction, representation, and recognition. To get good correspondence results, many approaches require large training data. But we focus on the dense point correspondence of 3D faces and its application on 3D face morphing and animation which require only two objects.

3. 3D Face Deformation Based on Thin Plate Spline

To get more accurate point matching result, the prototypic objects are generally transformed into a reference before alignment. There are rigid transformation, affine transformation, and nonaffine deformation. As the 3D faces have complex shape feature, it is difficult to find a rigid or affine transformation with good deformation results. The nonaffine transformation is considered as the proper mapping method. For the scanned 3D faces with high dimensional dense points, the data is too large to do a global transformation for all points. The alternative solution is to use subsampling sparse point sets. Here we use an

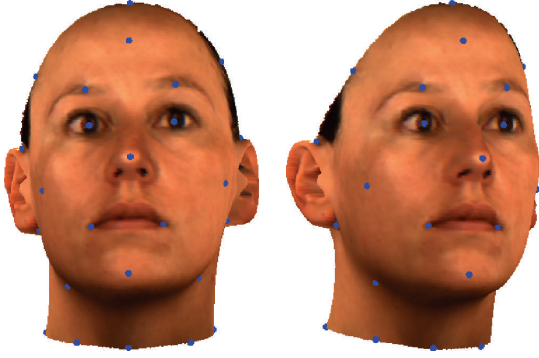


FIGURE 1: The landmarks placed on the 3D faces for TPS transformation using an interactive tool.

interactive tool to pick out 25 landmarks on the aligning 3D faces. Figure 1 shows the landmarks on the 3D faces. These landmarks are the main feature points that refer to the morphological properties of human face, and will be used as the controlling points to constraint the TPS deformation between 3D faces in our method.

It is frequent in spline theory to generate a smoothly interpolated mapping between two sets of landmark points. We adopt TPS to model the deformation of 3D faces. TPS was introduced by Harder and Desmarais [14], and Bookstein [15] firstly used TPS for medical image registration. TPS is a class of nonrigid spline mapping functions with desirable properties, such as globally smooth, and easily computable, and the most important is that TPS transformation can be separated into affine and nonaffine components. So TPS has been widely used in 2D image or 3D data registration for variety applications. The following gives the implementation of TPS transformation for 3D faces in detail.

The TPS transformation can be regard as a mapping from space R^3 to R^3 , so we denote TPS as $f : R^3 \mapsto R^3$. For the convenience of explication, we use F_1, F_2 that denote the source 3D face and destination 3D face for aligning. F_1, F_2 can be looked as two point sets hat have the following expression:

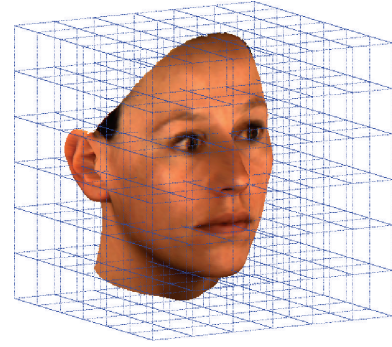
$$\begin{aligned} F_1 &= \{P_{1i} | P_{1i} = (x_{1i}, y_{1i}, z_{1i}), i = 1, \dots, N_1\}, \\ F_2 &= \{P_{2j} | P_{2j} = (x_{2j}, y_{2j}, z_{2j}), j = 1, \dots, N_2\}, \end{aligned} \quad (1)$$

where N_1 and N_2 are the points number of F_1 and F_2 such that $N_1 \leq N_2$. The landmark points sets of F_1 and F_2 are denoted as

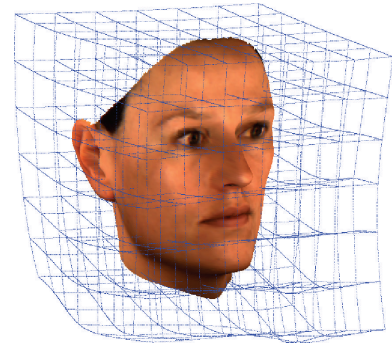
$$\begin{aligned} M_1 &= \{L_{1j} | L_{1j} = (x_{1j}^*, y_{1j}^*, z_{1j}^*), j = 1, \dots, M\} \\ M_2 &= \{L_{2j} | L_{2j} = (x_{2j}^*, y_{2j}^*, z_{2j}^*), j = 1, \dots, M\}, \end{aligned} \quad (2)$$

where M is the count of landmarks (here $M = 25$). These landmarks are the controlling points for TPS transformation, that is, TPS satisfies the following interpolation conditions at the landmark points:

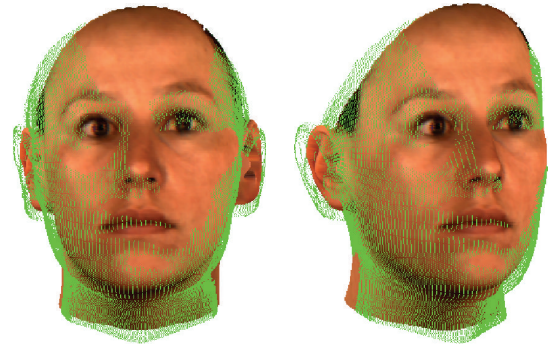
$$f(L_{1j}) = L_{2j}, \quad j = 1, \dots, M. \quad (3)$$



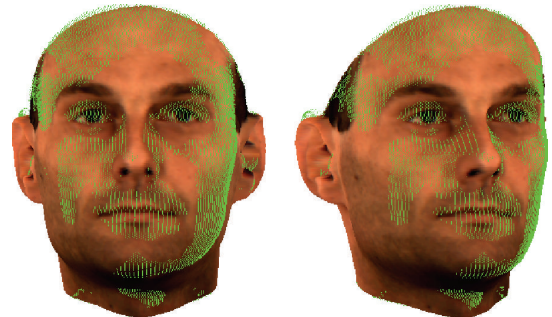
(a)



(b)



(c)



(d)

FIGURE 2: The TPS deformation of the source 3D face. The top one is the source 3D face with a standard partitioned cube. The second is the source 3D face deformed by TPS, and the distorted cube shows the spatial deformation of TPS. The third two are the images of the source 3D face comparing with the deformed source 3D face which displays as the sparse mesh. The bottom two are the images of the destination 3D face comparing with the deformed source 3D face.

At the same time, TPS is restricted by the blend smooth constraint, formed by the minimization of the following blending energy function, the sum of squares of all second-order partial derivatives:

$$E(f) = \iiint_{R^3} \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 + \left(\frac{\partial^2 f}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial xy} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial xz} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial yz} \right)^2 \right] dx dy dz. \quad (4)$$

It is proved that TPS can be decomposed by affine component and nonaffine component [15]. This fact is generally represented as the following formula:

$$f(P) = Pd + Kw, \quad (5)$$

where P is the point on the source 3D face F_1 and has the homogeneous coordinates $(1, x, y, z)$. d is a 4×4 affine transformation matrix. K named TPS kernel is an $1 \times M$ vector with the form $K = (K_1(P), \dots, K_M(P))$ such that $K_j(P) = \|P - L_{1j}\|$, $j = 1, \dots, M$. w is an $M \times 4$ warping coefficient matrix representing the nonaffine deformation.

To get TPS transformation, the matrices d and w must be determined. There are two solutions to this problem, the interpolating and noninterpolating methods. If TPS needs not be interpolated, that is, formula (3) is not strictly satisfied, the following energy function can be minimized to find the optimal answer:

$$E'(\lambda, w, d) = \frac{1}{M} \sum_{j=1}^M \|L_{2j} - f(L_{1j})\| + \lambda \cdot E(f), \quad (6)$$

where λ is the weight to control the smooth component, and for a fixed λ there will be a unique minimum for the energy function.

In the interpolating case, formula (3) is satisfied, putting (5) into (3), and confining w to nonaffine transformation, that is, $M_1^T w = 0$, it leads a direct solution for d and w formed by the following matrix relation:

$$\begin{bmatrix} w \\ d \end{bmatrix} = \begin{bmatrix} K' & M_1' \\ M_1'^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} M_2' \\ 0 \end{bmatrix}, \quad (7)$$

where M_1' and M_2' are $M \times 4$ matrix whose rows are the homogeneous coordinates of the landmark points belonging to M_1 and M_2 , respectively. K' is an $M \times M$ symmetry matrix which represents the spatial relation between the landmark points of the source 3D face and has the element k_{ij} with the following formation:

$$k_{ij} = \|L_i - L_j\|, \quad i = 1, \dots, M, \quad j = 1, \dots, M. \quad (8)$$

In our work, the landmarks placed on the source and target 3D faces are looked as the correspondent points with the same facial feature, hence the condition in (3) will be satisfied, and the interpolating method is adopted here to solve the TPS transformation. From (7) the matrices d and

w will be determined, and the source 3D face F_1 will be deformed by TPS transformation, we denote the deformed 3D face of F_1 as F_1' . Figure 2 shows the TPS deformation of the source 3D face and the deformed 3D face is compared with the source 3D face and the destination 3D face. It is proved that the deformed source 3D face is closer to the destination 3D face than the source 3D face, so it leads a more accurate points alignment. In the next section, the point-to-point correspondence between F_1' and F_2 will be done by a closest point matching process.

4. Dense Point Alignment by Closest Point Matching

Although the rigid transformation of ICP algorithm is not used in our method, we adopt the similar closest point matching schemes like ICP. That is, for each point on the deformed source 3D face F_1' , the closest point will be found on the destination 3D face F_2 . Before the closest point matching, the closest point criterion must be defined. ICP algorithm generally uses the distance between points or the distance between point and point set to define the closest point, and the distance refers to Euclidean distance. Here we define the closet point in the sense of the distance from a point to a point set. To the point P'_{1i} on F_1' , the correspondent point P_{2j} on F_2 is determined by the following minimum requirement:

$$P'_{2j} = \min_{j=1, \dots, N_2} \text{DIS}(P'_{1i}, P_{2j}), \quad (9)$$

where $\text{DIS}(\cdot)$ is a function defined to compute the distance between two points. As the deformation among 3D faces is a type of nonrigid transformation, the Euclidean distance used to determine the closest points in rigid transformation is not the proper method in nonrigid situation. Considering the modality of human face, the curvature is an important property interrelated to the local surface feature. Here the distance is defined as a weighted combination of Euclidean distance and the difference of the mean curvature of the points. The distance $\text{DIS}(P_1, P_2)$ of points P_1, P_2 has the following formation:

$$\text{DIS}(P_1, P_2) = \delta \cdot \|P_1 - P_2\| + (1 - \delta) \cdot |(MC(P_1) - MC(P_2))|, \quad (10)$$

where δ is the weight to balance the Euclidean distance and the curvature difference such that $0 \leq \delta \leq 1$. In the following experiments we set $\delta = 0.5$. $MC(\cdot)$ is the function to compute the mean curvature of the points on 3D faces.

Having determined the closest point matching criterion, for each point on F_1' , the closest point searching must be executed on the target 3D face F_2 . As the huge data of the source and target 3D faces, the whole closest points searching is a very time consuming procedure with computation $O(N_1 \times N_2)$. To get high point matching efficiency, we adopt the K dimensional binary search tree (KD-tree) technique in the point matching method. The KD-tree algorithm was introduced by Bentley [16] and has been widely utilized in the nearest neighbor searching [17]. It is a binary search

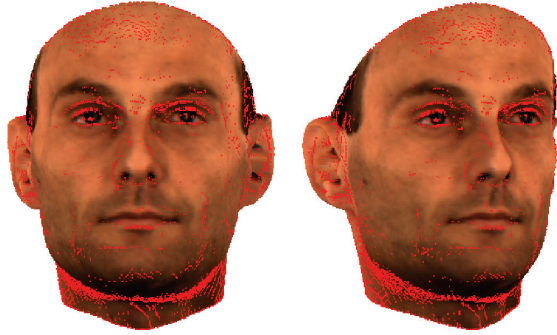


FIGURE 3: The collision points on the destination 3D face.

tree in which each node represents a partition of the k dimensional space. The root node represents the entire space, and the leaf nodes represent subspaces containing mutually exclusive small subsets of the relevant points. The space partitioning is carried out in a recursive binary fashion. The average performance of the KD-tree searching has complexity of $O(N_1 \times \log N_2)$.

The other obstacle has to be settled for the closest point matching is that the current method does not preserve one-to-one mapping. In fact, some points on the deformed 3D face F'_1 may be mapped onto the same point on the destination 3D face F_2 . We denote these points on F_2 as collision points which have more than one correspondent points on F'_1 . Generally the collision points are produced by the points of outliers or the points with local complex geometry feature. Considering the high resolution of 3D faces and the distribution of these collision points, the latter one is concerned with the main problem. The distribution of these collision points on the destination 3D face F_2 is shown in Figure 3. To eliminate these collision points, a revised point matching algorithm is proposed. The main idea of the method is to construct a distance list for every collision point, and only the point with minimum distance is regarded as the truly correspondent point. The following is the outline of the one-to-one point matching algorithm.

- (1) Create KD-tree for the destination 3D face F_2 .
- (2) For each point on the deformed source 3D face F'_1 , search its closest point on F_2 .
- (3) Detect the collision points on F_2 , if not exist, go to 6.
- (4) For each collision point P_{2j} , find the correspondent points on F'_1 reversely, denote the point with minimum distance as P'_{1i} , and record the correspondent pair points (P'_{1i}, P_{2j}) .
- (5) Remove the point P'_{1i} from F'_1 , delete the node P_{2j} from the KD-tree, then go to (2)
- (6) Record the remained correspondent pairs of points without collision.

By the revised closest point matching algorithm, the correspondent point searching procedure maintains one-to-one mapping, though more computation is required.

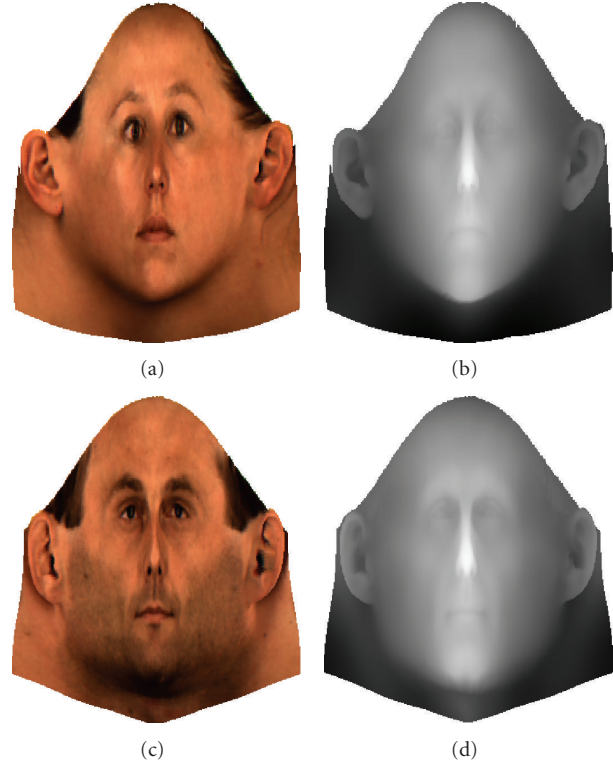


FIGURE 4: The texture and height mapping images for optical flow computation. The top two are the texture and height mapping images of the source 3D face. The bottom two are the texture and height mapping images of the destination 3D face.

5. Experimental Results of 3D Face Morphing and Animation

If the point-to-point correspondence of 3D faces is established, the direct application of the alignment is to create 3D face morphing and animation, which have wide applications in computer game, virtual reality, and animating actor in entertainment movies.

The scanned 3D faces we used come from MPI Face database [18] and BJUT-3D Face Database [19]. As the 3D facial scans have high resolution, which generally have more than 70 000 vertices and 140 000 triangles with texture information, the realistic animation results will be achieved if accurate point correspondence is obtained. Here we use the simple key-frames interpolation method to produce the face morphing and animation between the source and destination faces. The points on the key-frames 3D face are computed by linear interpolation between the correspondent points. The texture and the geometry normal of the correspondent points are interpolated at the same time.

The experiment of face morphing is implemented on two 3D faces selected from MPI face database, one face is female and the other is male. As the difference of the two faces is adequate to express variety of the human face modality, the nonrigid transformation is demanded to do with the deformation. The face animation is created on the same person's 3D faces with different expressions selected from

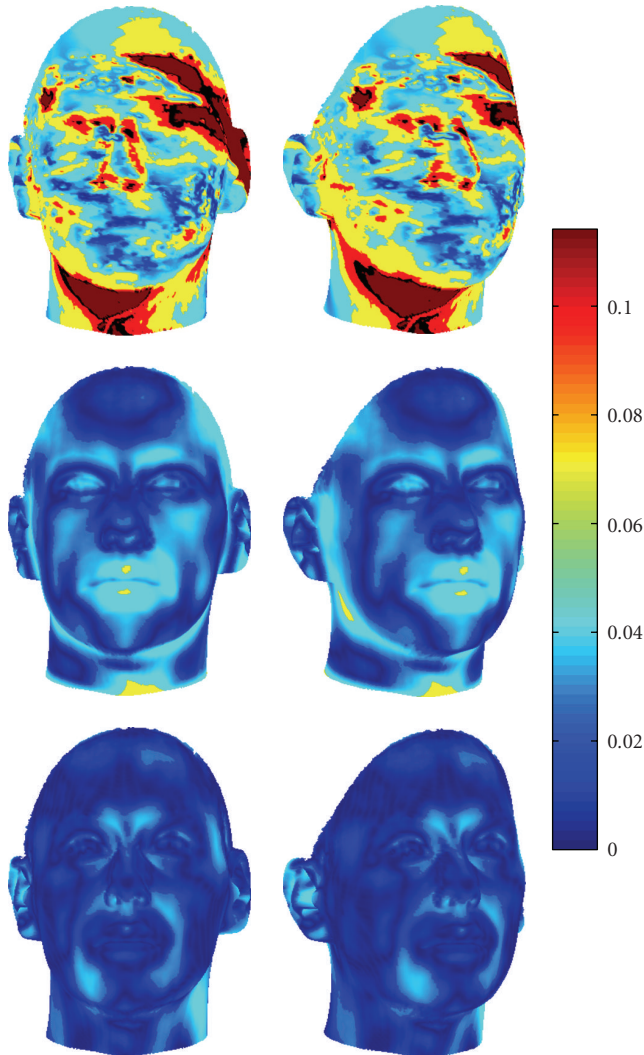


FIGURE 5: The distances of the correspondent points are visualized as colors on the source 3D face. The color of each point represents the distance from the point to its correspondent point with the color-mapping on the right. The top two are the results of the optical flow method. The middle two are the results of the ICP algorithm with rigid transformation. The bottom two are the results of the TPS method.

BJUT-3D Face Database. The sequence of key-frames of the face morphing and face animation is shown in Figure 7. On the whole, the vision reality of the morphing and animation is satisfied, though the local areas with relative complex shape feature and the areas with missing points as the scanning reason are not looking good, such as the areas of mandible and ears.

To compare our TPS method with the original ICP algorithm [7] and the optical flow method [9], the MPI source 3D face is aligned to the target 3D face using these three methods, respectively. To compute the point correspondence by the optical flow method, the source and target 3D faces are spread into texture and height mapping images (shown in Figure 4) by cylinder coordinate

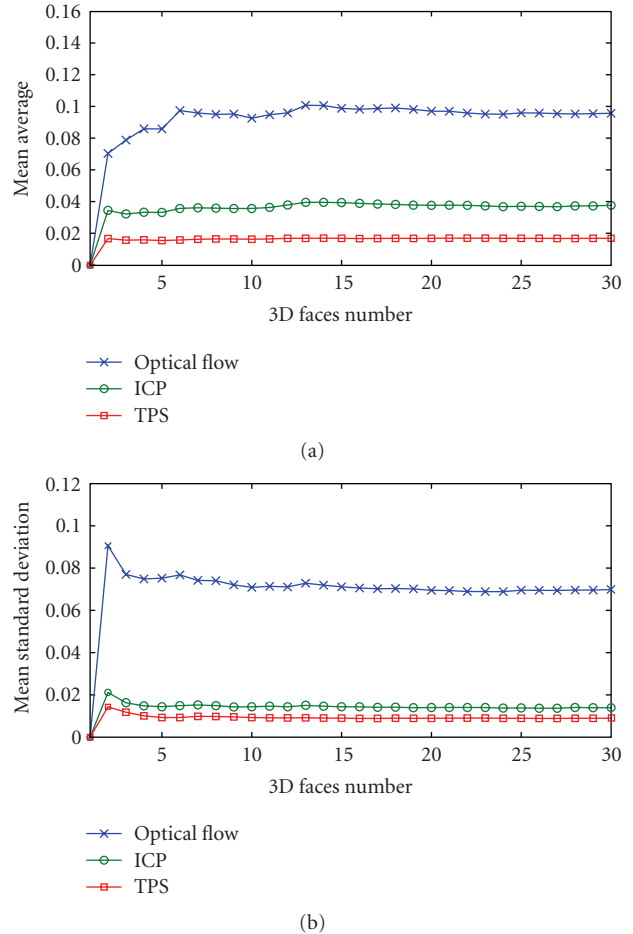


FIGURE 6: The trend of the mean average and standard deviation of the distances between the correspondent points of the 3D faces in the aligning set with 3D faces number increasing. The top is the mean average of the distances. The bottom is the mean standard deviation of the distances.

transformation. Then the facial texture and height mapping images are aligned by an optical flow algorithm, here we adopt the optical flow algorithm proposed by Horn and Schunck [20]. Finally the point correspondence of 3D faces is obtained from the alignment of 2D images by the reversed cylinder coordinate transformation. In ICP and TPS methods, the source 3D face is transformed by rigid transformation and TPS deformation, respectively. Then using the proposed closest point searching method, the two transformed faces are aligned with the destination 3D face. To evaluate the alignment results of these three methods, the average and standard deviations of the distances between the correspondent points on the source and destination 3D face are computed respectively.

The results of these three methods are shown in Table 1. It is denoted that all the vertices of the 3D faces are standardized into $[0, 1]$ interval before the experiment. The distances of correspondent points of these three methods are also visualized on the source 3D face (shown in Figure 5). The average and standard deviations of the distances and



FIGURE 7: The sequence of 3D face morphing and animation. The left column and the right column are the source 3D faces and the destination 3D faces. The top two rows are the middle frames morphing a female 3D face to a male 3D face selected from MPI 3D face database. The middle two rows are the animation sequence of a person from the neutral state to an aspiratory action state. The bottom two rows are the animation of a person from the neutral expression to smile expression. The 3D faces in the bottom four rows are selected from BJUT-3D Face Database.

its visualization in Figure 5 reveal that the TPS method has the best point matching accuracy, while the optical flow method performs poorly in dense points alignment, and the ICP is in-between of the former two methods. The optical

flow is generally used in perception of the movement of objects in video sequence [21]. When the difference between the facial images is too large to satisfy the continualness requirement of adjoining frame images, the optical flow

TABLE 1: The average and standard deviation of the distances between the correspondent points on the source 3D face and destination 3D face.

| | Optical flow | ICP | TPS |
|-------------------------------------|--------------|---------|---------|
| Average of the distances | 0.05683 | 0.01673 | 0.00804 |
| Standard deviation of the distances | 0.03840 | 0.01069 | 0.00637 |

computation will fail with obvious error. It is the main reason for referring to the poor results of the optical flow method. In fact, the nonrigid transformation is more suitable for 3D faces deformation than rigid transformation, so that the TPS method has the better results than ICP algorithm.

To examine the stability of the TPS method, we selected 30 3D faces from BJUT-3D Face Database as an aligning set. The dense point alignment is implemented on the aligning set using the above three methods. The experiment is done with the 3D faces number of the aligning set increasing, that is, the 3D faces are added into the aligning set gradually. At first, the aligning set composes of two 3D faces, then 3D faces are added into one by one, until all 30 face are added. At the same time, the mean average and standard deviations of the correspondent points distances of the 3D faces in the aligning set are computed. Figure 6 shows the change of the mean average and standard deviations with the increasing of 3D faces number respected to the optical flow method, ICP algorithm and TPS method. The experimental results show that the mean average distance and its standard deviations of these three methods are all converging toward a stable value, and TPS method has better stability and correspondence accuracy than the ICP algorithm and the optical flow method.

6. Conclusion

In this paper, we describe a new dense point-to-point alignment method and apply it on scanned 3D faces. In the method, TPS is adopted to model the deformation of 3D faces, and a closest point matching algorithm is proposed to search the correspondent points and simultaneously guarantees the alignment one-to-one mapping. To reduce the closest points searching time and get good point matching accuracy, a KD-tree technique and a user-defined distance function which considers the points local curvature are integrated with the point matching algorithm. The dense point alignment is used in 3D faces morphing and animation by key-frames interpolation and gets satisfied realistic visual results. Contrasting with ICP algorithm and the optical flow method, the error analysis on the selected pair of MPI 3D faces and the experiment on 30 BJUT 3D faces prove that our method is efficient for dense point correspondence. Furthermore, the method does not require large facial database and can easily extend to other dense objects.

In our work, the landmarks of 3D faces are picked up by an interactive tool, though the manual marking procedure is simple, and taking little time, it limits the method apply in many areas, such as realtime application and the large quantity of objects situation. So the future work firstly focus on the fully automatic point matching algorithm. The intuitively thought is to find the suitable automatic feature detection method, but it is another challenging problem in pattern recognition and computer vision. The additional points to be improved of this work include refining the aligning accuracy by exploring proper representation of the local geometry feature, constructing the whole head model with hair to get more natural looking, and making practical applications.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 60736008 and no. 60872127) and the Postdoctoral Science Foundation of China (Grant no. 20080430316). The 3D facial scans were provided by the Max-Planck Institute for Biological Cybernetics in Tuebingen, Germany and the Multimedia and Intelligent Software Technology Beijing Municipal Key Laboratory of Beijing University of Technology in Beijing, China.

References

- [1] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, pp. 187–194, Los Angeles, Calif, USA, August 1999.
- [2] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.
- [3] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.
- [4] M. A. Audette, F. P. Ferrie, and T. M. Peters, "An algorithmic overview of surface registration techniques for medical imaging," *Medical Image Analysis*, vol. 4, no. 3, pp. 201–217, 2000.
- [5] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [6] R. Wan and M. Li, "An overview of medical image registration," in *Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '03)*, p. 385, Xi'an, China, September 2003.
- [7] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [8] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling*, pp. 145–152, Quebec, Canada, May-June 2001.
- [9] T. Vetter and V. Blanz, "Estimating coloured 3D face models from single images: an example based approach," in *Proceedings of the 5th European Conference on Computer Vision (ECCV '98)*, vol. 2, pp. 499–513, Freiburg, Germany, June 1998.

- [10] F. Steinke, B. Schölkopf, and V. Blanz, "Learning dense 3D correspondence," in *Advances in Neural Information Processing Systems 19*, pp. 1313–1320, MIT Press, Cambridge, Mass, USA, 2007.
- [11] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.
- [12] V. Jain and H. Zhang, "Robust 3D shape correspondence in the spectral domain," in *Proceedings of IEEE International Conference on Shape Modeling and Applications (SMI '06)*, pp. 118–129, Matsushima, Japan, June 2006.
- [13] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 75–84, Orlando, Fla, USA, July 1998.
- [14] R. L. Harder and R. N. Desmarais, "Interpolation using surface splines," *Journal of Aircraft*, vol. 9, no. 2, pp. 189–191, 1972.
- [15] F. L. Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1992.
- [16] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [17] M. Greenspan and M. Yurick, "Approximate K-D tree search for efficient ICP," in *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling (3DIM '03)*, pp. 442–448, Banff, Canada, October 2003.
- [18] N. F. Troje and H. H. Bühlhoff, "Face recognition under varying poses: the role of texture and shape," *Vision Research*, vol. 36, no. 12, pp. 1761–1771, 1996.
- [19] Y. Hu, B. Yin, Y. Sun, and S. Cheng, "3D face animation based on morphable model," *Journal of Information and Computational Science*, vol. 2, no. 1, pp. 35–39, 2005.
- [20] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.
- [21] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

