Research Journal of Applied Sciences, Engineering and Technology 5(5): 1804-1810, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: August 09, 2012 Accepted: September 03, 2012 Published: February 11, 2013

A 3D Facial Expression Tracking Method Using Piecewise Deformations

[1]Jing Chi, [2, 3]Xiaoming Wu and [1]Shanshan Gao
[1]Department of Computer Science and Technology, Shandong University of Finance and Economics, No. 7366 Erhuan East Road, Lixia District, Ji'nan 250014, Shandong, China
[2]Shandong Computer Science Center, Ji'nan 250014, China
[3]Shandong Provincial key Laboratory of Computer Network, Ji'nan 250014, China

Abstract: We present a new fast method for 3D facial expression tracking based on piecewise non-rigid deformations. Our method takes as input a video-rate sequence of face meshes that record the shape and time-varying expressions of a human face, and deforms a source mesh to match each input mesh to output a new mesh sequence with the same connectivity that reflects the facial shape and expressional variations. In mesh matching, we automatically segment the source mesh and estimate a non-rigid transformation for each segment to approximate the input mesh closely. Piecewise non-rigid transformation significantly reduces computational complexity and improves tracking speed because it greatly decreases the unknowns to be estimated. Our method can also achieve desired tracking accuracy because segmentation can be adjusted automatically and flexibly to approximate arbitrary deformations on the input mesh. Experiments demonstrate the efficiency of our method.

Keywords: Mesh matching, non-rigid transformation, segmentation, time-varying expressions

INTRODUCTION

3D facial expression tracking is one of the most interesting yet difficult problems in computer graphics. It plays an important role in many applications such as synthetic character animation, expression recognition, face modeling, virtual reality, and so on.

Many techniques for 3D facial shape acquisition in real time have been explored recently. These techniques can acquire sequence of non-rigidly deforming 3D face meshes at video rate. These meshes record the shape and time-varying expressions of a human face. Such a mesh sequence is inherently unstructured and uncompressed because at each time frame the acquired mesh has different geometry and connectivity. It is difficult to establish intra-frame correspondence on such a mesh sequence, and consequently, to track the subtle expressional variations and reanimate the acquired expressions. Therefore, it is needed to reconstruct these meshes to generate new meshes with the same connectivity, and integrally represent the new meshes into a single deformable mesh model. The new mesh sequence reflects the time-varying expressions, and the single deformable mesh model supports further processing such as expression editing, surface deformation analysis, whole filling, and so on. In this study, we address the problem of tracking the acquired video-rate face meshes to simulate dynamic facial expressions.

Huang *et al*. (2004), Amberg *et al*. (2007) and Blanz *et al*. (2007) all use high-resolution deformable models to track facial expressions. Amberg *et al*. (2007) propose a non-rigid Iterative Closest Point (ICP) method for surface registration in expression tracking by introducing adjustable stiffness parameters into the traditional ICP framework. The method computes an affine transformation for per vertex of the high-resolution face model so as to deform the face model to accurately simulate expressional variations. Huang *et al*. (2004) fit a multi-resolution face model to a sequence of face point clouds. They track global rigid deformations on the coarse level of the face model and local non-rigid deformations on the fine level. The non-rigid registration integrates an implicit shape representation and B-spline based Free Form Deformation (FFD), which may increase computational complexity.

Bickel *et al*. (2007), Bickel *et al*. (2008), Ma *et al*. (2008), Furukawa and Ponce (2009) and Huang *et al*. (2011) focus on tracking fine-scale facial details such as wrinkles and furrows. Bickel *et al*. (2007) and Bickel *et al*. (2008) use a video sequence and motion capture markers of an actor's performance to track medium-scale expression wrinkles. Furukawa and Ponce (2009) model non-rigid tangential deformation on tangent

Corresponding Author: Jing Chi, Department of Computer Science and Technology, Shandong University of Finance and Economics, No. 7366 Erhuan East Road, Lixia District, Ji'nan 250014, Shandong, China

planes of the source mesh to track facial wrinkles on the cheeks and neck. Huang *et al*. (2011) leverage high-fidelity motion capture data and high-resolution face scans for tracking facial wrinkles and fine-scale stretching and compression. All these methods should place markers on actor's face, and too many markers may make actor uncomfortable.

Wand *et al*. (2007), Süßmuth *et al*. (2008) and Wand *et al*. (2009) consider the spatial and temporal coherence of the face mesh sequence in expression tracking. Süßmuth *et al*. (2008) compute an implicit function in R4 to approximate the time-space surface of the real-time point clouds. The method can get coherent meshes approximating the input data at arbitrary time instances. Wand *et al*. (2007) and Wand *et al*. (2009) automatically compute a fitting shape and its non-rigid motion from the time-varying point clouds. The computations of these methods are relatively complex since they perform on both space domain and time domain.

Zhang *et al*. (2004), Borshukov *et al*. (2005), Dornaika and Ahlberg (2006) and Wang *et al*. (2008) utilize optical flow to guide automatic expression tracking. Zhang *et al*. (2004) compute optical flow from 2D image sequences, and then uses optical flow to automatically constraint matching between the deformable model and the time-varying point clouds. Borshukov *et al*. (2005) use optical flow to track each vertex's motion in 2D and use 3D stereo to triangulate 3D positions of these vertices. Estimation of optical flow is complex and not robust in some case, e.g., for those points having no texture information in 2D image, their motions in 3D space cannot be constrained.

In this study, we present a piecewise deformation-based method to track facial shape and subtle expressional variations quickly. We use a deformable mesh model to match each mesh in the input sequence acquired at video rate. We introduce segmentation idea in mesh matching by representing the deformations between two meshes as piecewise non-rigid transformations. We automatically segment the deformable mesh using a variation of ICP framework, and compute a non-rigid transformation for each segment that will deform the deformable mesh to approximate the input mesh accurately. Piecewise non-rigid transformation greatly decreases the unknowns to be optimized for mesh matching because it computes a non-rigid transformation for each segment, not for each vertex of the deformable mesh as do many existing methods. Moreover, the number of segments is small when there are only coarse deformations between two meshes. Therefore, our method can track time-varying expressions quickly. Additionally, segmentation can be adjusted automatically and flexibly to approximate arbitrary deformations on the input mesh, so our method can achieve desired tracking accuracy by increasing segments.

GENERAL SCHEME

Our method takes as input a mesh sequence acquired at video rate. We assume that the input mesh sequence consists of M frames. At the *m*-th frame, the mesh is represented as $T_m = \{V_m\}$, V_m is vertex set. Let $S = \{V\}$ be the source mesh, with vertex set $V = \{v_i\}$. The source mesh can either be automatically obtained from the first frame of the input sequence or from a user-defined mesh model. Our goal is to deform the source mesh to match through the input sequence.

At each frame, we automatically segment the source mesh and compute an affine transformation for each segment to match the non-rigid deformations on the input mesh. Take the *m*-th frame for example, in order to match the input mesh T_m, we assume that the source mesh S is segmented into N segments represented as $S = \{S_1, S_2,\ldots, S_N\}$, and the computed affine transformations are 4×4 matrices represented as $\{D_1, D_2,\ldots, D_N\}$. The source vertices in one segment have the same affine transformation. Applying these piecewise affine transformations to the source mesh, we can get a new mesh as follows:

$$S' = \{D_i v_i\},\ D_i \in \{D_1, D_2, ..., D_N\}$$

The mesh S' approximates T_m closely and maintains the connectivity of S. Therefore, by deforming the source mesh S to match each input mesh, we can output a new sequence of meshes with the same connectivity. The output sequence approximates the non-rigid deformation dynamics of the input sequence and reflects the time-varying expressions.

MESH SEGMENTATION

Basic idea: When we consider matching between the source mesh and the input mesh, we expect that the non-rigid deformations between two meshes can be approximated by piecewise non-rigid transformations. The basis of piecewise non-rigid transformations can be established because it spans the domain of all possible non-rigid deformations and in the least compact case each source vertex could has its own affine transformation that will transform the source mesh onto the input mesh closely.

At each frame, we determine a segmentation of the source mesh together with a non-rigid transformation for each segment. Specifically, we first compute an

affine transformation on the source mesh using an improved ICP framework, and then any source vertices that transform further than a given threshold are rejected from the segment. Once a segment is computed we iterate the process until all source vertices have been segmented.

Non-rigid transformation estimation: In order to estimate non-rigid transformations with high accuracy, we improve the traditional ICP framework by introducing a new optimization criterion that is a variant of the mesh matching criterion from Chi and Zhang (2011). The new criterion considers not only distance constraint but also normal constraint in closest point search. Specifically, to achieve optimal mesh matching, each source vertex, deformed with the affine transformation, should approach the input mesh as close as possible, as well as have the same normal direction as its corresponding point on the input mesh as soon as possible. Therefore, the new criterion consists of two constraint terms.
The first term represented as:

$$E_1(\{D\}) = \sum_{v_i \in V} \omega_i \left\| Dv_i - q_i \right\|^2 \quad (1)$$

is called closest-point term. It measures the distance between the deformed source vertex and the input mesh.

where,
D = The unknown affine transformation to be estimated
Dv_i = The new position of v_i after transformation
q_i = The closest point on the input mesh from point Dv_i
ω_i = A weight factor that will be set to 0 where no corresponding closest point is found for v_i

The second term represented as:

$$E_2(\{D\}) = \sum_{v_i \in V} \mu_i Agl^2(N_{Dv_i}, N_{q_i}) \quad (2)$$

is called normal-keeping term. It measures the directional difference between normal's on the deformed source vertex and its corresponding closet point on the input mesh.

Where,
Dv_i & q_i = Defined as in Eq. (1)
N_{Dvi} = The vertex normal on Dvi
N_{qi} = The surface normal on q_i
Agl() = The angle between two normal vectors
μ_i = A weight facto that takes the same value with ω_i in Eq. (1)

The new criterion is a weighted sum of Eq. (1) and (2) represented as follows:

$$E(\{D\}) = \alpha E_1 + \beta E_2 \quad (3)$$

where, α and β are weights to blend two constraint terms.

The segmentation steps: We give the segmentation process now. First, we compute a uniform affine transformation D for all the source vertices by minimizing Eq. (3) and then only consider those vertices that transform to points within a given threshold, i.e., classify those vertices that satisfy the following equation:

$$S_n = \{v_i \mid \left\| Dv_i - q_i \right\|^2 < Dis\} \quad (4)$$

in to a segment S_n.

where,
Dis = A user-defined threshold

Once a segment is generated, we repeat the above process for all unmatched vertices in S to get another new segment S_n, i.e., Let $S = S - S_n$, We compute a new subset S_n by minimizing Eq. (3) on S and using Eq. (4) for selection. The process is iterated until all source vertices have been matched.

METHOD IMPLEMENTATION

The implementation details of our method are discussed in this section. If the source mesh is automatically obtained from the first frame of the input sequence, we will directly use it to track through the subsequent input meshes. If the source mesh is a user-defined mesh, then it may be very different from the input meshes in facial shapes. In this case, to get more ideal tracking results, we first match the source mesh to the first frame using traditional non-rigid ICP method with some manual aid, and then, use the deformed source mesh to track through the rest of the input sequence automatically.

In order to use as few segments as possible to approximate the input mesh with high accuracy, we should include as many vertices as possible in each segment to estimate an affine transformation. Therefore, we give a large Dis initially for estimate each segment in our method. Once the affine transformation is computed by optimizing Eq. (3) and the subset S_n is determined with the large Dis, we decrease Dis and perform Eq. (3) and (4) on S_n again until either Dis is as small as the user-defined threshold or the number of vertices in S_n is less than 3. The case that the number of vertices is less than 3 occurs when the user-defined threshold is too small, which can be solved by increasing the user-defined threshold.

After deforming the source mesh with the computed segmentation and piecewise affine transformations, we will project each deformed source vertex along its normal onto the input mesh to get the final mesh. The projection can:

- Further improve the matching accuracy
- Efficiently compensate possible discrepancy between different segments since all the source vertices are located on the input mesh

EXPERIMENTS AND RESULTS

Our method is implemented using C++ under the Windows XP environment. We performed our method on a variety of face mesh sequences acquired at video rate. All these experiments run on a 2.27 GHz Core i3 processor. We present some results of our experiments in this section.

Fig.1 shows results of tracking a mesh sequence acquired at 25 fps. Adjacent frames in the sequence are very close spatially and temporally, so the deformations between adjacent frames are very small. Therefore, fewer segments of the source mesh are enough to approximate these non-rigid deformations. As shown in Fig. 1(a) is the source mesh automatically obtained from the first frame, (b) is the second frame, and (c) is another frame in the sequence. (d) and (f) are respectively the results of matching frame (b) and (c) with the source mesh in tracking process. (e) and (g) show the segmentations of the source mesh in (d) and (f). Where, there are 3 segments in (e), and 5 segments in (g). It can be seen that the intra-frame non-rigid deformations in the input sequence are well approximated by the source mesh with fewer segments.

Figure 2 shows results of tracking another mesh sequence with our method. The intra-frame deformations in this sequence are finer than that of the sequence in Fig. 1, so more segments are needed to track this sequence. In this experiment, we use a user-defined mesh as the source mesh and match it to the first frame at first. The initial matching result is shown in Fig. 2 (a), (b) and (c) are two frames in the input sequence. (d) Shows the segmentation of the source mesh and the result of matching (b). Here, the source mesh is segmented into 7 segments. It can be seen

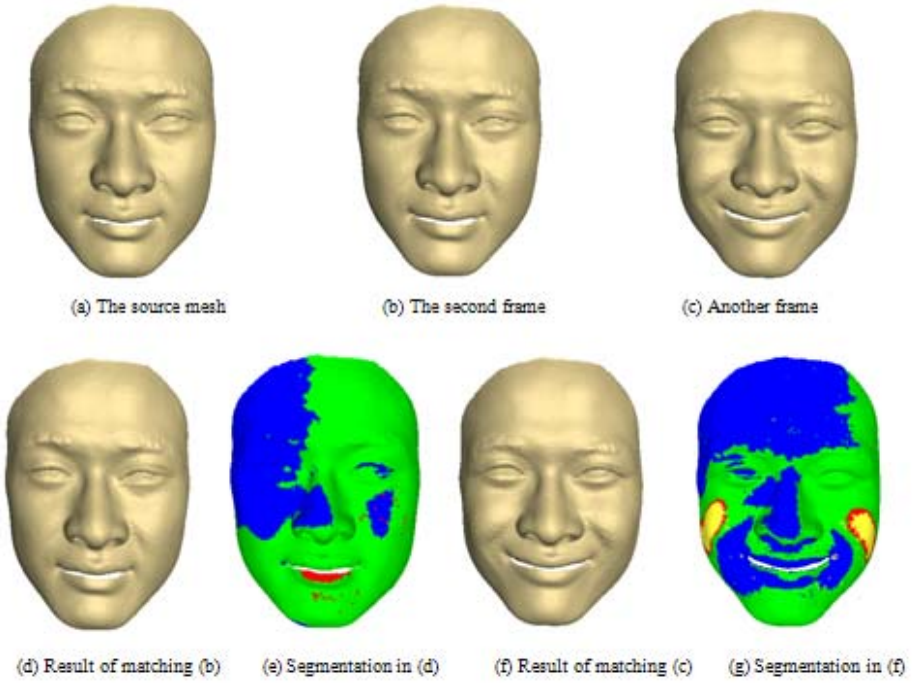

Fig. 1: The results of tracking a mesh sequence with our new method

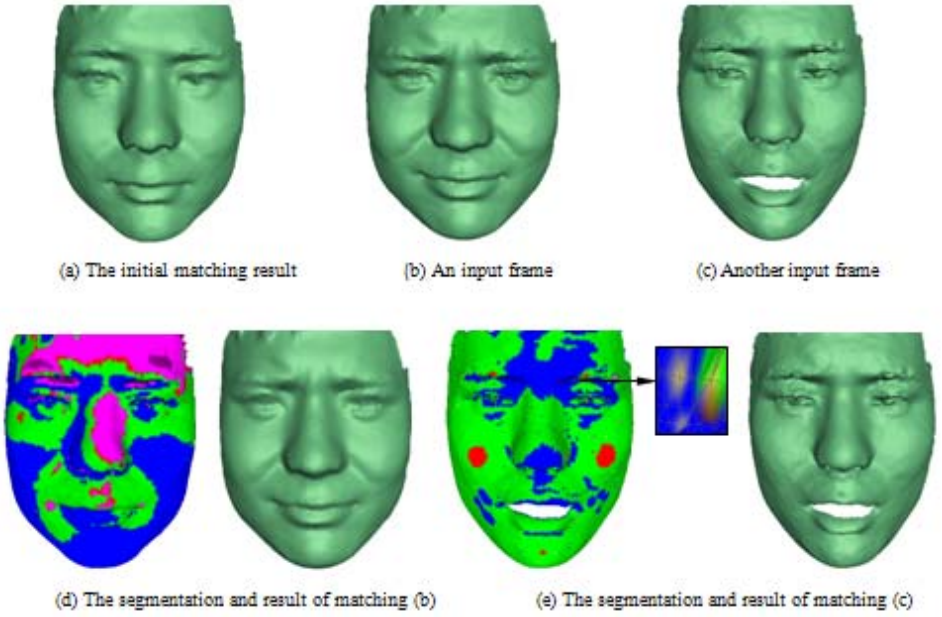

Fig. 2: The results of tracking another mesh sequence with our new method

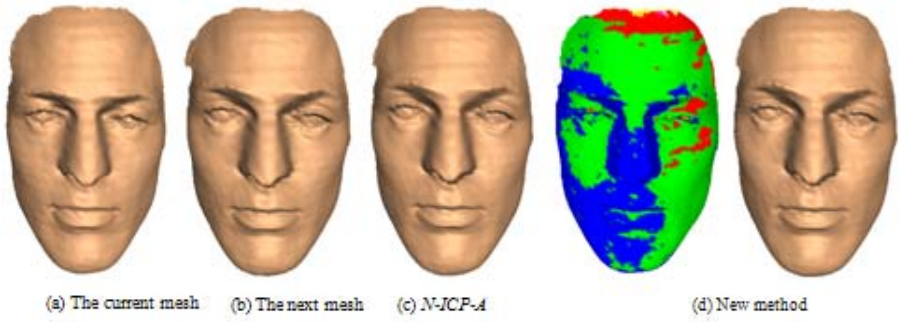

Fig. 3: The matching of two adjacent meshes

that small segments appear around the regions of nose, eye, and mouth to approximate the fine deformations, and large segments appear other regions to approximate coarse deformations. (e) Shows the segmentation of the source mesh and the result of matching (c). Here, the source mesh is segmented into 10 segments. Some segments are very small to approximate the very fine deformations, e.g., the small segments around the eyebrow as shown in rectangle.

To evaluate the efficiency of our method, we compare our method with *N-ICP-A* method (Amberg *et al*., 2007) in matching accuracy and running time. *N-ICP-A* method computes an affine transformation for per source vertex to deform the source mesh towards the input mesh. As shown in Fig. 3 (a), (b) are two adjacent frames in an input sequence. We use (a) as the source mesh to match mesh (b). (c) is the matching result obtained with *N-ICP-A* method. (d) is the matching result and segmentation of the source mesh obtained with our new method, here, the number of segments is 5. The geometric information about meshes (a) and (b) is listed in Table 1 and comparisons of matching accuracy and running time are listed in Table 2. We use the average squared Euclidean distance of all corresponding points on the source mesh after

Table 1: Numbers of vertices and triangles of meshes in Fig. 3

Mesh	Vertices	Triangles
Fig. 3 (a)	10275	20037
Fig. 3 (b)	10580	20679

Table 2: The efficiency comparison of N-ICP-A and our method in Fig. 3

Method	N-ICP-A	New method
Accuracy	6.58E-7	1.62E-6
Time	21.57s	12.97s

Table 3: Numbers of vertices and triangles of meshes in Fig. 2

Mesh	Vertices	Triangles
Fig. 2 (a)	9962	19459
Fig. 2 (b)	10053	19668
Fig. 2 (c)	9936	19374

Table 4: The efficiency comparison of *N-ICP-A* and our method in Fig. 2

	Matching mesh (b)		Matching mesh (c)	
Method	*N-ICP-A*	New method	*N-ICP-A*	New method
Accuracy	6.95E-7	1.29E-6	1.56E-6	1.66E-6
Time	35.32s	16.12s	27.95s	15.19s

deformation and the input mesh to measure the matching accuracy. It can be seen from Table 2 that, our new method runs much faster than *N-ICP-A* method meanwhile maintains desired accuracy.

We compared the efficiency for all the experiments. Table 3 and 4 list respectively the mesh information and the comparison result in Fig. 2. It can be seen from these comparisons that, our method significantly improve the speed with little decrease in accuracy compared with *N-ICP-A* method. The accuracy of our method is still high and perfectly acceptable in practice. Moreover, the accuracy of our method can be further improved flexibly by increasing segments of the source mesh. Therefore, our method can flexibly adjust to different application needs of speed and accuracy.

CONCLUSION

In this study, we present a new method for tracking video-rate facial mesh sequence. The method uses a source mesh to match through the whole input sequence. Segmentation and piecewise non-rigid transformations are introduced in each matching. Piecewise non-rigid transformations greatly decrease the unknowns to be computed, so the method has high speed. Additionally, the segmentation of the source mesh can be adjusted automatically and flexibly. When there are only coarse deformations between two meshes, the segments could be large, so it will approximate the deformations quickly. When there are fine deformations between two meshes, it can approximate the deformations with desired accuracy by increasing segments.

We treat each input frame as a separate mesh for matching in our method, which may result in accumulative error in some cases. Taking into consideration the spatial and temporal coherence in the input sequence and keeping high speed and accuracy is our future study.

ACKNOWLEDGMENT

This study is supported by National Nature Science Foundation under Grant 60903109, Nature Science Foundation of Shandong Province under Grant ZR2010FQ031, and Jinan Youth Star Program under Grant 201101-0113.

REFERENCES

Amberg, B., S. Romdhani and T. Vetter, 2007. Optimal step nonrigid ICP algorithms for surface registration. Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, Minnesota, USA, Jun 18-23, pp: 1-8.

Bickel, B., M. Lang, M. Botsch, M. Otaduy and M. Gross, 2008. Pose-space animation and transfer of facial details. ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Dublin, Ireland, Jul 7-9, pp: 57-66.

Bickel, B., M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister and M. Gross, 2007. Multi-scalecapture of facial geometry and motion. ACM Trans. Graph., 26(3): Article 33.

Blanz, V., K. Scherbaum and H. Seidel, 2007. Fitting a morphable model to 3d scans of faces. Proceeding of IEEE Conference on Computer Vision, Rio de Janeiro, Brazil, Oct. 14-21, pp: 1-8.

Borshukov, G., D. Piponi, O. Larsen, J.P. Lewis, and C. Tempelaar-Lietz, 2005. Universal capture-image-based facial animation for the matrix reloaded. ACM SIGGRAPH 2005 Courses, Los Angeles, USA, Jul 31-Aug. 4, pp: 16.

Chi, J. and C.M. Zhang, 2011. Automated capture of real-time 3D facial geometry and motion. Comput-Aid. Design Appl., 8(6): 859-871.

Dornaika, F. and J. Ahlberg, 2006. Fitting 3d face models for tracking and active appearance model training. Image Vision Comput., 24(9): 1010-1024.

Furukawa, Y. and J. Ponce, 2009. Dense 3d motion capture for human faces. Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Florida, USA, Jun 20-25, pp: 1674-1681.

Huang, H., J. Chai, X. Tong and H. Wu, 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. Proceeding of ACM SIGGRAPH'11, Vancouver, BC, Canada, Aug. 7-11, 30(4): Article 74.

Huang, X., S. Zhang, Y. Wang, D. Metaxas and D. Samaras, 2004. A hierarchical framework for high resolution facial expression tracking. Proceeding of IEEE Workshop on Articulated and Nonrigid Motion (ANM'04) in Conjunction with CVPR'04, Washington D.C., USA, Jun. 2, pp: 22-29.

Ma, W.C., A. Jones, J.Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung and P. Debevec, 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. ACM Trans. Graph., 27(5): 1-10.

Süßmuth, J., M. Winter and G. Greiner, 2008. Reconstructing animated meshes from time-varying point clouds. Comput. Graph. Forum, 27(5): 1469-1476.

Wand, M., P. Jenke, Q. Huang, M. Bokeloh, L. guibas and A. Schilling, 2007. Reconstruction of deforming geometry from time-varying point clouds. Proceedings of the 5th Euro Graphics Symposium on Geometry Processing, Barcelona, Spain, July 4-6, pp: 49-58.

Wand, M., B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.P. Seidel and A. Schilling, 2009. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. ACM Trans. Graph., 28(2): Article 15.

Wang, Y., M. Gupta, S. Zhang, S. Wang, X.F. Gu, D. Samaras and P.S. Huang, 2008. High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps. Int. J. Comput. Vision, 76(3): 283-300.

Zhang, L., N. Snavely, B. Curless and S.M. Seitz, 2004. Spacetime faces: High resolution capture for modeling and animation. ACM Trans. Graph., 23(3): 548-558.