ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

3D Mesh Simplification for Deformable Human Body Mesh Using Deformation Saliency

Tianhao ZHAO
R301, SHB, CUHK
Shatin, N.T.
999077, Hong Kong
thzhao@ee.cuhk.edu.hk

King Ngi NGAN
R304, SHB, CUHK
Shatin, N.T.
999077, Hong Kong
knngan@ee.cuhk.edu.hk

Songnan LI
R301, SHB, CUHK
Shatin, N.T.
999077, Hong Kong
snli@ee.cuhk.edu.hk

ABSTRACT

3D mesh of human body is the foundation of many hot research topics, such as 3D body pose tracking. In this topic, the deformation of the human body mesh has to be taken into account because of various poses of the human body. Considering the time cost of the body deformation, however, it's impractical to adopt a high resolution body mesh generated from scanning systems for the real-time tracking. Mesh simplification is a solution to reduce the size of body meshes and accelerate the deformation process.
In this paper, we propose a mesh simplification algorithm using deformation saliency for such deformable human body meshes. This algorithm is based on quadric edge contraction. The deformation saliency is computed from a set of meshes with various poses. With this saliency, our algorithm can simplify the 3D mesh non-uniformly. Experiment shows that using our algorithm can improve the accuracy of body pose simulation in the simplified resolution compared to using classical quadric edge contraction methods.

Keywords

Mesh simplification, Deformable human body mesh, Deformation saliency.

1 INTRODUCTION

Nowadays, 3D human body tracking is a hot research topic [Bog15, Baa13, Wei12, Gan10]. In this topic, 3D body mesh is a basic structure. Generally, the input of 3D body tracking is a sequence of depth images captured by depth cameras. A body mesh in an initial pose is selected as a template mesh. The template mesh is deformed to different poses according to the real-time input frames. The body mesh is defined as the combination of a point cloud and a set of triangulated faces. The faces cover all the points and there is no overlay between any two faces. Typically the 3D mesh is obtained from a multiple depth-camera scanning system or a laser scanning system. The mesh generated by these scanning systems is in very high resolution, containing tens of thousands of vertices and faces, such as the CAESAR dataset. Nevertheless, 3D human body tracking is required to be real-time, so the body mesh in such high resolution is not suitable for the real-time tracking. Reducing the mesh resolution, i.e., the number of vertices and faces of the mesh is necessary.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In this paper, we focus on the mesh simplification for such deformable human body object. We adopt the deformable human body model proposed by [Ang05]. The body meshes of SCAPE model are from the CAESAR dataset. The mesh standing in the "A" pose (shown in Fig.1) is selected as the template mesh and other meshes contain various poses of the same person. All the meshes have the same number of vertices, which have been registered. In different meshes, the topologies of triangulated faces are also the same. The human body is partitioned into 16 parts, containing head, chest, torso, arms, hands, legs and feet. The motion model of SCAPE describes two kinds of deformations of human bodies: the rigid transformation and non-rigid deformation. The rigid transformation matrices show the global transformation of an integral body part. For all the vertices in the same body part, their rigid transformation matrices of a certain pose are the same. The non-rigid deformation matrices indicate the change of muscle, skin and ligament in different poses. These matrices are unique for each face. Apparently, there are much more non-rigid deformations at joints or muscular regions than other body regions. Therefore, in simplification more vertices in these special regions should be preserved to simulate the non-rigid deformation more accurately.

However, there is a drawback of the traditional mesh simplification algorithms when they are applied to such deformable meshes: they always simplify a 3D mesh

uniformly or based on geometric features, which means that more vertices will be kept in some geometry-salient regions, like fingers and face of the human. This may lead to very few vertices remaining in the highly non-rigid deforming regions. Hence, our purpose is to distinguish the highly non-rigid deforming regions, and keep more vertices in these regions when simplifying the body mesh.

Based on the quadric edge contraction (Qslim) [Gar97], we developed a mesh simplification algorithm guided by deformation saliency. In our algorithm, we compute the deformation saliency for every vertex in the template mesh from the SCAPE pose database. Besides the deformation saliency, we also introduce a distance balancing item to avoid the over-contraction. Our method can simplify meshes region-specifically according to the non-rigid deformation levels. Experimental result shows that using our simplification method, the deformed result is more accurate than that using the Qslim.

This paper is organized as following. Section 2 introduces the related work. Section 3 discusses our methodology. Section 4 analyzes experimental results and compares the proposed method with the prior studies. Conclusion is drawn in Section 5.

2 RELATED WORK

Mesh simplification aims to reduce the number of vertices of a 3D mesh, meanwhile preserving the shape of the mesh as accurately as possible. To this end, the Quadric Edge Collapse Decimation (Qslim) was proposed by [Gar97]. In their paper, each vertex of the mesh was associated with a quadric error matrix, which was defined as the squared distances from this vertex to the planes of its adjacent triangulated faces. Every edge had a contraction cost based on its potential contraction-target vertex and the quadric error matrices of its two endpoints. Iteratively the edges with the minimum contraction costs were contracted to the optimal target vertex which could minimize the contraction costs. The contraction costs for all related vertices would be updated in each iteration.

One trend to improve the mesh simplification method is to reduce the running time. Using GPU is a common way to speed up the computation. [Sho13] proposed a CPU-GPU combined algorithm, which has the lowest computational cost compared to all the other methods just running on CPU. Some methods like [Cam13] considered both accuracy and simplicity to obtain the optimal simplification result. Another multilevel refinement method was proposed in [Mor14], which combined a Laplacian flow to the high resolution mesh. This method can locate the most appropriate regions to be contracted in different refinement levels, which took into account both accuracy and speed.

Another trend to improve the simplification is to keep more details. Mesh features are used widely in the related work. More vertices in the regions with prominent features will be preserved. For example, mesh curvature is a common feature used in the prior studies [All03, Lee05, Wan11, Yao15] . [All03] introduced curvature directions to represent the intrinsic anisotropy of the mesh geometry. In [Lee05], mesh feature was defined as a center-surround operator on Gaussian-weighted mean curvatures. In [Wan11], by measuring the curvature, the authors proposed a method to perform coarse simplification in flat regions and fine simplification near creases and corners respectively. [Yao15] adopted the discrete curvature to modify the Qslim method.

Besides curvature, there are many other saliencies used for mesh simplification. [Tol08] introduced acceleration and deceleration of vertices as saliency for simplification of dynamic meshes. [Zha12] identified visually important regions by points sampling method to keep the mesh saliency. [Pey14] reduced the number of vertices by Possion disk sampling based on features such as sharp edges or corners, and re-meshed vertices along the detected feature lines. In [Pel14], Pellerin et al. applied Centroidal Voronoi optimization to simplify the mesh and merged features for the mesh with complex contacts. A B-rep feature-based model was proposed in [Kim14]. [Ng14] developed a method called half-edge collapsed scheme. They identified valid decimated edges by the length of the edges and the difference between every two adjacent faces. Another distinctive method was proposed in [Van15], which adopted outgoing radiance functions of the mesh surface as the mesh feature. In [Ďur15], shape diameter function was adopted as mesh saliency to extract skeleton, which can also be used in mesh simplification.

3 METHODOLOGY

Our purpose is to simplify the human body mesh while preserving more vertices at joints and other regions with prominent non-rigid deformations. Given a set of meshes M of various poses and a template mesh T of the SCAPE data set, we compute the deformation saliency along with a balancing weight for each vertex. Then we iteratively contract a valid edge on the template mesh to generate a new mesh T' in lower resolution. The indices of the preserved vertices are recorded. For the other meshes, the vertices with the same indices will be kept, so that the topologies of the triangulated faces of these meshes are the same as that of the simplified template mesh T'.

3.1 Deformation saliency

We define the deformation saliency as the Euclidean distance of the corresponding vertices between the

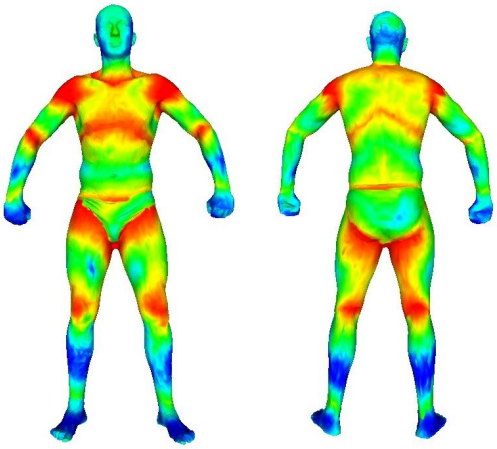

Figure 1: The heat maps of saliency values on the template mesh. Red means the highest value, and blue means the lowest value

rigidly transformed template mesh and the mesh in the target pose, which is also described by the non-rigid deformation in the SCAPE model. First we estimate the rigid transformation between the template mesh T and each mesh M_X of pose X in the pose dataset, which means to calculate the integral rotation matrix for each body part between mesh T and M_X in the global coordinates. In the SCAPE database, vertices have been registered across different meshes and partitioned into 16 body parts. For each body part k, the rotation matrix and translation vector are denoted as R_k and t_k, respectively. To estimate R_k and t_k, we minimize the following energy function:

$$E_{transform} = \underset{R_k,t_k}{\operatorname{argmin}} \sum_{v_i^T, v_i^{M_X} \in \text{part}(k)} \|R_k v_i^T + t_k - v_i^{M_X}\|_2^2 \tag{1}$$

In this least square function, v_i^T is the ith vertex in the mesh T and $v_i^{M_X}$ is the ith vertex in the mesh M_X.

By vectorizing R_k and t_k, Eq. (1) can be turned into a system of linear equations, which can be solved analytically. With the rigid transformation, we can transform each body part of the template mesh via this equation:

$$v_i^{T_X} = R_k v_i^T + t_k, v_i \in \text{part}(k) \tag{2}$$

where T_X is the rigidly transformed template mesh corresponding to the mesh M_X.

After generating the rigidly transformed meshes T_X in all poses X, we calculate the Euclidean distance error between every vertex of T_X and its corresponding vertex of M_X. For each vertex, we add up its distance errors calculated from all the poses X. Then we normalize the total error and take it as the deformation saliency for each vertex. Fig.1 shows the saliency values around the human body.

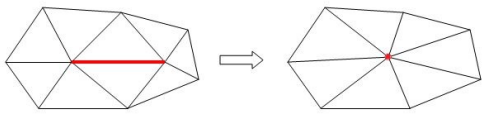

Figure 2: The red edge denotes the edge with the minimum deformation weight and contraction cost; it is contracted to a new vertex (the red point) which can minimize the contraction cost; other related vertices are re-connected to the new vertex.

The vertex with high value of deformation saliency means there is prominent non-rigid deformation, so in the simplification its possibility to be preserved should be larger than other vertices. As the Qslim algorithm illustrated [Gar97], every vertex holds a quadric matrix Q, which represents the entire set of planes adjacent to this vertex. For an edge (v_1, v_2), the contraction cost associated to its potential target vertex v_t is defined as:

$$E_{contract} = v_t^T (Q_1 + Q_2) v_t \tag{3}$$

In this equation, Q_1 and Q_2 are the quadric matrices of v_1 and v_2, respectively. v_t is the optimal target vertex which can minimize the contraction cost $E_{contract}$. The contraction cost indicates the sum of squared distances from v_t to the plane set represented by Q_1 and Q_2.

Using a couple of weights w_1, w_2 to represent the deformation saliencies of v_1, v_2, we update the edge contraction cost as:

$$E_{contract} = \frac{w_1 + w_2}{2} \cdot v_t^T (Q_1 + Q_2) v_t \tag{4}$$

The deformation weight of an edge is defined as the average weight of its two attached endpoints. However, an extreme result is that too many edges may be contracted in the regions where vertices have low deformation weights. In this case, very long edges may be generated; meanwhile few edges are contracted in the regions where vertices have high deformation weights. To solve this dilemma, we introduce a balancing weight for each edge, which is equal to the length of the edge (v_1, v_2). The effect of this balancing weight is to reduce the contracting priority of long edges. By denoting this balancing weight as d, the edge contraction cost is rewritten as:

$$E_{contract} = \frac{d(w_1 + w_2)}{2} \cdot v_t^T (Q_1 + Q_2) v_t \tag{5}$$

Iteratively, the edge with the least contraction cost is contracted, generating the simplified template mesh T'.

Fig.2 illustrates edge contraction in a single iteration, and Fig.3 compares the simplified results with and without the balancing item.

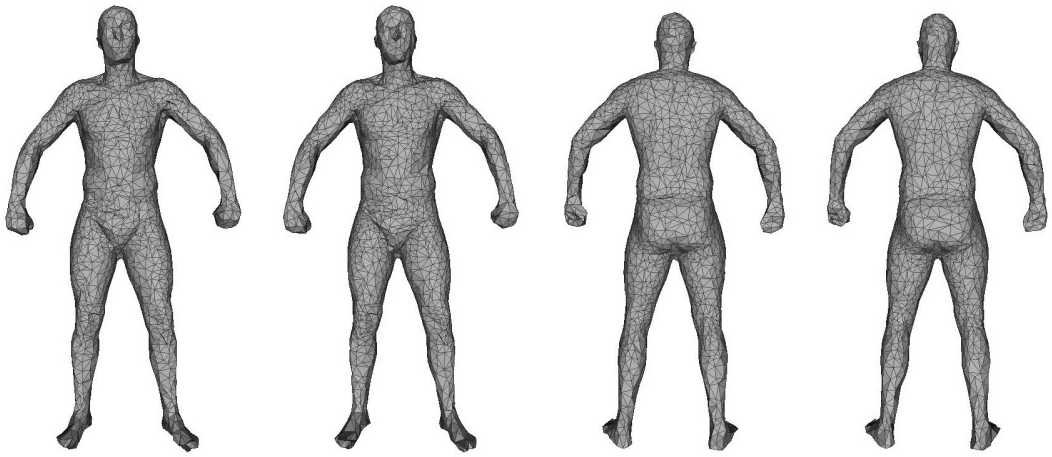

Figure 3: From left to right: front side of the simplified template mesh with and without balancing item; back side of the simplified template mesh with and without balancing item. Edges of the meshes are shown explicitly. With the balancing item, the variance of edge length is 1.53e-04; without it, the variance of edge length is 3.08e-02. This proves that the balancing item can prevent too long edges occurring effectively.

3.2 Greedy least distance mapping

According to the SCAPE, to build the body deformation model, vertex partitions and face topologies of all the meshes should be the same, and all the meshes should be registered. So we use a greedy least distance mapping from T' to T to keep vertices registered across different simplified meshes.

Initially, we define the simplified candidates as all vertices of T' and the original candidates as all vertices of T. Iteratively, between the two candidates we find the mapping vertex pair with the least Euclidean distance, record the vertex index of this pair, and take them out from the candidates. All vertices of T' are mapped to distinctive vertices of T. With the mapping record, we can simplify other meshes via preserving the vertices with the same indices as in the record. As a consequence, all simplified meshes still contain the registered vertices and the same body partition.

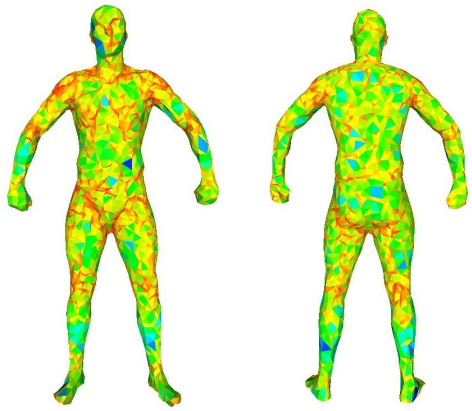

Figure 4: The area heat maps of the simplified template mesh using our method (2500 vertices, 5000 faces).

4 EXPERIMENTS

We conducted the experiments on a subset of the SCAPE database. We chose 70 body meshes in different poses of the same person as the pose dataset. Each of them has 12500 vertices and 25000 triangulated faces. In the 70 meshes, the first mesh standing in the "A" pose was selected as the template mesh. Based on the pose set, deformation saliency was computed. With our method and the compared methods, the template mesh and other meshes were simplified. The number of faces and vertices were reduced from 25000 to 5000 and from 12500 to 2500, respectively. Then the template mesh was deformed to other poses. The errors between the deformed template mesh and the mesh of the target pose were computed to evaluate the proposed methods.

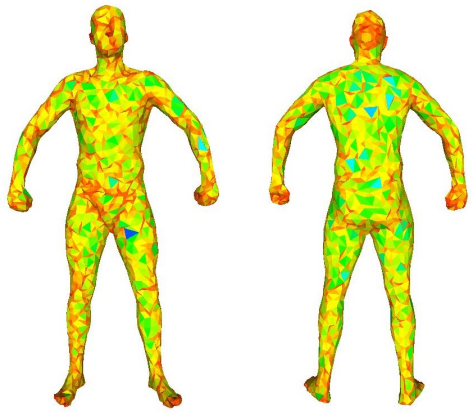

Figure 5: The area heat maps of the simplified template mesh using Qslim (2500 vertices, 5000 faces).

4.1 Area heat map of the simplified result

The visualized comparison between the results of the proposed method and the Qslim method is drawn in Fig.4 and Fig.5, which are referred to as the heat maps of the area of the triangulated faces.

In both figures, the faces are colorized according to the size of their area: smaller faces have warmer color, while larger faces have cooler color. That is to say, red regions have the highest vertex density, and blue regions have the lowest vertex density. Yellow and green regions have median vertex density.

By observing the two heat maps, we can find that vertex densities are quite different in the same region across the results produced by different methods. We know the head, hands, and feet typically have more geometric details. Therefore, the Qslim method preserves more vertices in these regions. On the contrary, it is clear that we preserve more vertices at shoulders, knees, elbows, even in chest and thighs; meanwhile we preserve fewer vertices in the regions like hands, head and feet. This means that we have more vertices to simulate the non-rigid deformation at joints and muscular regions.

4.2 Deformation errors

In this section, we measure the deformation errors between the deformed template mesh and the mesh of the target pose. Less deformation error means that the simplification method is more suitable for such deformable human body mesh to change the body pose. The detail of the body pose deformation can be referred to [Ang05]. Besides the 70 meshes simplified by our method and the Qslim, we also simplify 10 meshes using the method provided in the open-source CGAL library for comparison. The CGAL (http://www.cgal.org/) is a widely-used library of geometry algorithms. We introduce two criteria to measure the deformation errors:

1. Vertex-to-face error (v2f)

The vertex-to-face error measures the average squared distance between the deformed template mesh and the simplified mesh of the target pose, from a vertex to the closest face. It is also adopted by the Qslim method [Gar97]. The vertex-to-face error $E_i = (M_i, M_n)$ between the deformed template M_i and ground-truth M_n is defined as:

$$E_i = \frac{1}{|X_i| + |X_n|} \left(\sum_{v \in X_i} d_1^2(v, M_n) + \sum_{v \in X_n} d_1^2(v, M_i) \right) \quad (6)$$

where X_i, X_n are the point clouds in the mesh M_i and M_n respectively. The distance function $d_1(v,M) = \min_{p \in M} \|v - p\|$ is the minimum Euclidean distance from the vertex v to the closest face p on the mesh M. The measurement unit of this error is $meter^2$.

Method	v2f error	v2v error
Ours(70 poses)	1.08e-09	3.83e-05
Qslim(70 poses)	1.12e-09	4.14e-05
CGAL(10 poses)	1.10e-09	3.93e-05

Table 1: The average deformation errors of our method, Qslim and CGAL.

2. Vertex-to-vertex error (v2v)

The vertex-to-vertex error measures the average squared distance from a vertex of the deformed template mesh to the closest vertex of the simplified mesh of the target pose. This metric is adopted in [Bog15]. Based on the previous notations, the vertex-to-vertex error $E_i = (M_i, M_n)$ is defined as:

$$E_i = \frac{1}{|X_i|} \sum_{v \in X_i} d_2^2(v, M_n) \quad (7)$$

The distance function $d_2(v,M) = \min_{u \in M} \|v - u\|$ is the minimum Euclidean distance from the vertex v to the closest vertex u on the mesh M. The measurement unit is also $meter^2$.

The comparison results between our method and other methods are shown in Table 1. The average errors show that our method can reduce the deformation errors in both metrics. The reduction of the v2v error is more significant, which approaches about 8 percent compared to the Qslim. The results indicate that using our method, the simplified human mesh can be deformed more accurately than using the Qslim and CGAL.

Fig.6 shows some simplified results of different identities with the same pose, and Fig.7 shows the pose deformation results after simplification for several poses.

5 CONCLUSION

In this paper, we propose a mesh simplification method using deformation saliency for 3D human pose deforming. Based on SCAPE dataset, we compute the vertex distance error caused by non-rigid deformation for each vertex as the deformation saliency. In the template body mesh we contract the edges with the lowest saliency in prior. We also add a balancing weight to avoid generating too long edges caused by over-contraction. Our method shows better performance of body pose deformation compared to the Qslim and CGAL algorithms. Similarly, our method can be also applied to simplifying the meshes of other deformable objects for pose deformation.

6 REFERENCES

[Bog15] Bogo F, Black M J, Loper M, et al. Detailed Full-Body Reconstructions of Moving People from Monocular RGB-D Sequences. Proceedings of the IEEE International Conference on Computer Vision, pp.2300-2308, 2015.

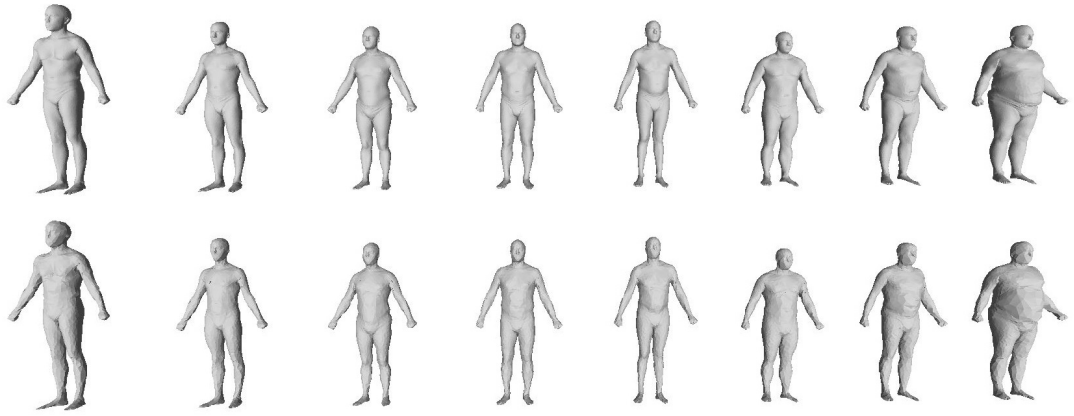

Figure 6: The first row shows several examples of the high resolution meshes from the SCAPE database (12500 vertices and 25000 faces); the second row shows our simplified meshes (2500 vertices and 5000 faces).

[Baa13] Baak A, Muller M, Bharaj G, et al. A data-driven approach for real-time full body pose reconstruction from a depth camera. Consumer Depth Cameras for Computer Vision, pp.71-98, 2013.

[Wei12] Wei X, Zhang P, Chai J. Accurate realtime full-body motion capture using a single depth camera. ACM Trans. Graph., 31(6), pp.188, 2012.

[Gan10] Ganapathi V, Plagemann C, Koller D, et al. Real time motion capture using a single time-of-flight camera. Computer Vision and Pattern Recognition, pp.755-762, 2010.

[Ang05] Anguelov D, Srinivasan P, Koller D, et al. SCAPE: shape completion and animation of people. ACM Trans. Graph., 24(3), pp.408-416, 2005.

[Gar97] Garland M, Heckbert P S. Surface simplification using quadric error metrics. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp.209-216, 1997.

[Sho13] Shontz S M, Nistor D M. CPU-GPU algorithms for triangular surface mesh simplification. Proceedings of the 21st international meshing roundtable, pp.475-492, 2013.

[Cam13] Campomanes-Alvarez B R, Cordon O, Damas S. Evolutionary multi-objective optimization for mesh simplification of 3D open models. 20(4), pp.375-390, 2013.

[Mor14] Morigi S, Rucci M. Multilevel mesh simplification. The Visual Computer, 30(5), pp.479-492, 2014.

[All03] Alliez P, Cohen-Steiner D, Devillers O, et al. Anisotropic polygonal remeshing. ACM Trans. Graph., 22(3), pp.485-493, 2003.

[Lee05] Lee C H, Varshney A, Jacobs D W. Mesh saliency. ACM Trans. Graph., 24(3), pp.659-666, 2005.

[Wan11] Wang J, Wang L, Li J, et al. A feature preserved mesh simplification algorithm. Journal of Engineering and Computer Innovations, 6, pp.98-105, 2011.

[Yao15] Yao L, Huang S, Xu H, et al. Quadratic Error Metric Mesh Simplification Algorithm Based on Discrete Curvature. Mathematical Problems in Engineering, 2015.

[Tol08] Tolgay A. Animated mesh simplification based on saliency metrics. bIlkent university, 2008.

[Pey14] Peyrot J L, Payan F, Antonini M. Aliasing-free simplification of surface meshes. International Conference on Image Processing, pp.4677-4681, 2014.

[Pel14] Pellerin J, Levy B, Caumon G, et al. Automatic surface remeshing of 3D structural models at specified resolution: A method based on Voronoi diagrams. Computers and Geosciences, 62, pp.103-116, 2014.

[Kim14] Kim B C, Mun D. Feature-based simplification of boundary representation models using sequential iterative volume decomposition. Computers and Graphics, 38, pp.97-107, 2014.

[Ng14] Ng K W, Low Z W. Simplification of 3D Triangular Mesh for Level of Detail Computation. Computer Graphics, Imaging and Visualization, pp.11-16, 2014.

[Van15] Vanhoey K, Sauvage B, Kraemer P, et al. Simplification of meshes with digitized radiance. The Visual Computer, 31(6-8), pp.1011-1021, 2015.

[Zha12] Zhao Y, Liu Y, Song R, et al. A saliency detection based method for 3d surface simplification. Acoustics, Speech and Signal Processing, pp.889-892, 2012.

[Ďur15] Ďurikovič R, Madaras M. Controllable Skeleton-Sheets Representation Via Shape Diameter Function. Mathematical Progress in Expressive Image Synthesis II, pp.79-90, 2015.

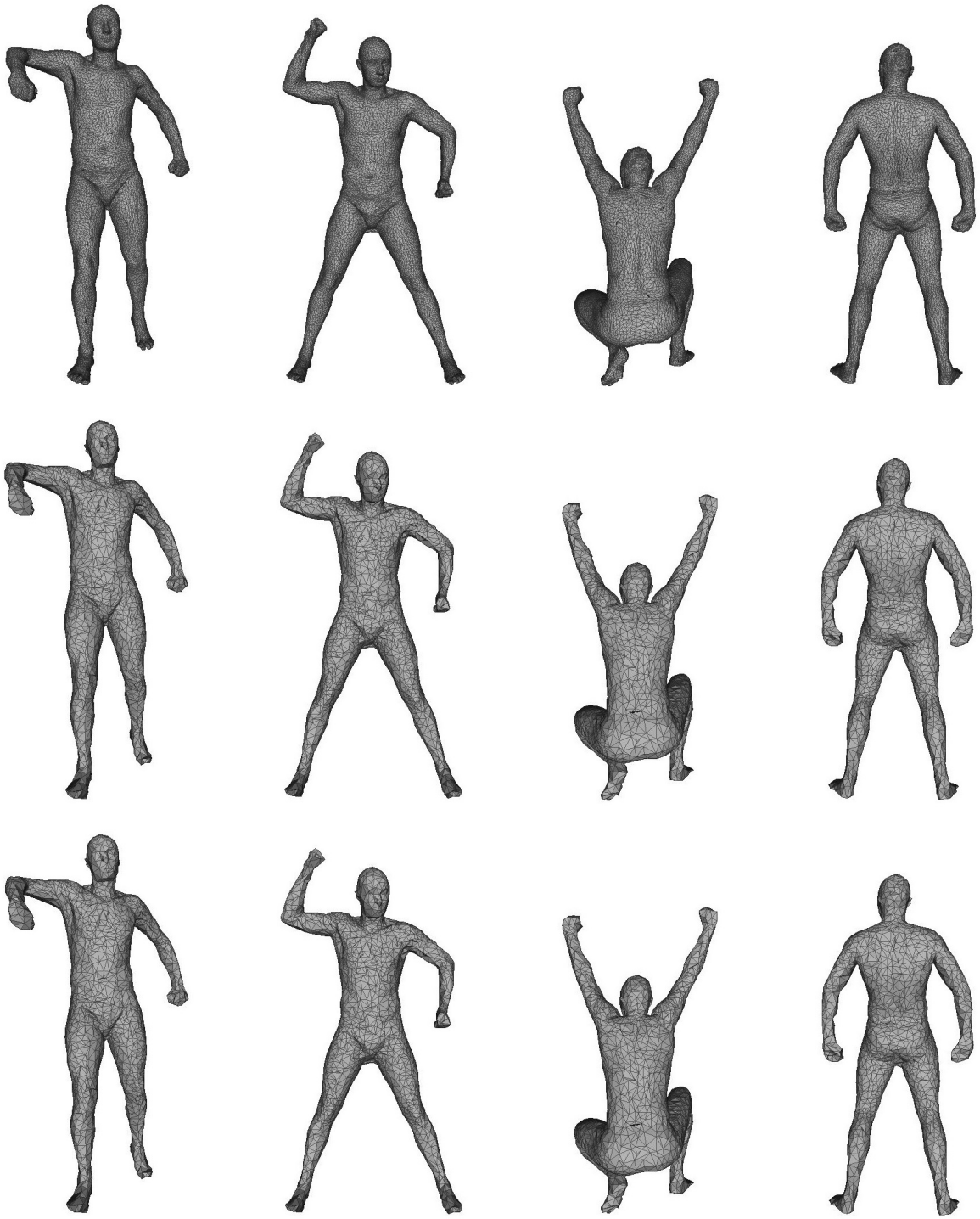

Figure 7: The first row shows some original meshes of different target poses; the second row shows the simplified results of these target poses; the third row shows the deformed results from the simplified template mesh to these target poses.