

Deformable Mesh Model for Complex Multi-Object 3D Motion Estimation from Multi-Viewpoint Video

Shohei NOBUHARA Takashi MATSUYAMA

Graduate School of Informatics, Kyoto University
Sakyo, Kyoto, 606-8501, Japan
{nob,tm}@vision.kuee.kyoto-u.ac.jp

Abstract

We propose a new algorithm using deformable mesh model for complex 3D motion estimation of multiple objects from multi-viewpoint video. In this paper, we define “complex motion” as motion which includes global change of the object shape topology. In complex motion, a part of the object may touch the other parts. To manage this effect, we introduce (1) “repulsive force” into deformable mesh model for simple motion estimation which integrates texture and silhouette information into unified computation scheme, and (2) efficient collision detection algorithm for deformable mesh model. Our deformable mesh model with repulsive force keeps hidden, collided surfaces to be touched each other, and gives dense, non-rigid complex 3D motion of the object. Some experimental results show that our deformation model can estimate motions of multiple objects and the object’s motion with time-varying global topology, and gives topologically-consistent mesh models which can be compressed efficiently by conventional inter-frame 3D data compression algorithms and be used for 3D motion analysis.

1. Introduction

3D motion estimation of complex human actions is an essential requirement for 3D archive and analysis of human activities, e.g., intangible cultural assets, and 3D data compression based on inter-frame per-vertex correspondences[3][6]. In this paper, we propose a new algorithm using deformable mesh model for 3D motion estimation of multiple, complex human actions from multi-viewpoint video. One of the most important advantages of deformable mesh model is integration of multiple estimation cues such as photo-consistency, silhouette boundary, smoothness and continuity of the object surface and motion. That is, once we can represent each of

cues as a force working on vertices of the mesh model and let them move vertices so as to satisfy the constraint corresponding to the estimation cue, we can deform the mesh to be “balanced” form which satisfies multiple estimation constraints. In recent years, some algorithms for static 3D shape estimation which integrates texture and silhouette constraints by deformable mesh model have been proposed[4][7][8]. We can use deformable mesh model not only for static 3D shape estimation but dynamic 3D shape and motion estimation[9]. Suppose we have a 3D mesh representing of the object shape at frame t given by conventional algorithm[7][8][11], and deform it so as to be the object shape at next frame $t + 1$ based on photometric, silhouette, and continuity constraints. Here, we have two mesh models of object shape at both t and $t + 1$, and we know how each vertex have been deformed from t to $t + 1$. This means that the *inter-frame deformation* of deformable mesh model gives dense and non-rigid 3D motion of the object. However, this inter-frame mesh deformation approach cannot cope with *complex* motion which includes global topological change of the object shape. That is, in complex motion, a part of the object may touch the other parts, – e.g., shaking hands by two person, or touching the arm on the waist. This is because the method defines vertex forces only based on geodesic distance between vertices of the mesh and ignores Euclidean distance. In simple motion, geodesic and Euclidean distance between vertices can be assumed to be similar, so each vertex can care only for its geodesical neighbors and it is quite easy to find vertices in geodesical vicinity from mesh model. However, in complex motion including global change of object topology, this assumption will not be true where two surface is about to collide. Hence, in this paper, we focus on *complex* 3D motion estimation including changing of apparent global shape topology, and propose an efficient algorithm to find collided surfaces.

This paper is organized as follows. We discuss related

work in Section 2, and review basic deformation model briefly in Section 3. Then we introduce our approach for 3D complex motion estimation in Section 4. We show experimental results in Section 5 and conclude this paper in Section 6.

2. Related Work

For human action understanding, many papers have been proposed which use human model given a priori and change its shape according to some visual information. For example, Heap[5] proposed a human hand tracking system from camera images using a given deformable hand model, and Bottino[1] tracked 3D human action based on multi-viewpoint silhouettes. However, they only estimate the motions of model parts, and cannot estimate dense motion of the object. Toward this problem, Plänkers[10] utilized a soft object model and deform it so as to fit the observed 2D silhouette. Vedula[12] proposed a method which estimates 3D shape and dense motion from multi-viewpoint images simultaneously without any prerequisite model, but it cannot work effectively for texture-less surfaces since it is a kind of space-carving method.

Compared to these researches, mesh deformation approach proposed in [9] does not require any special human model, and can work even for texture-less surfaces by integrating multiple estimation cues. However, it can estimate simple motion only, and cannot estimate complex motion which includes topological changing of the object shape.

3. Basic Deformable Mesh Model

We use heterogeneous deformation method proposed in [9] as the basic deformation model, and we review it briefly in this section.

The basic idea of 3D motion estimation by heterogeneous deformation is an inter-frame deformation of a 3D mesh model. Suppose we have a mesh model representing the object shape at frame t . If we can deform it to be the object shape at $t + 1$ by estimating the translation vectors of each vertices composing the mesh, we can obtain the 3D shape at $t + 1$ and dense, per-vertex motion between t and $t + 1$. We estimate the translation vectors based on videos observed from multiple viewpoints, but as is well known, we cannot expect that all the points on the object surface can be observed from cameras, nor observed images of the object have identifiable prominent textures all over the scene. Hence, we have to employ some constraints on translation vectors, i.e., object shape and motion, to make the estimation be stable. We represent each constraint as a force working at each vertex and compute how each vertex moves under these forces.

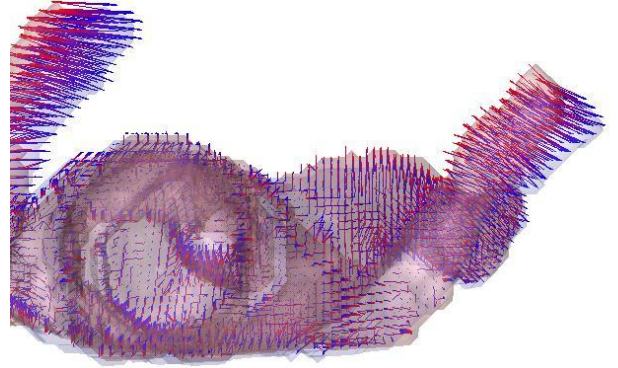


Figure 1. Roughly estimated motion flow lines

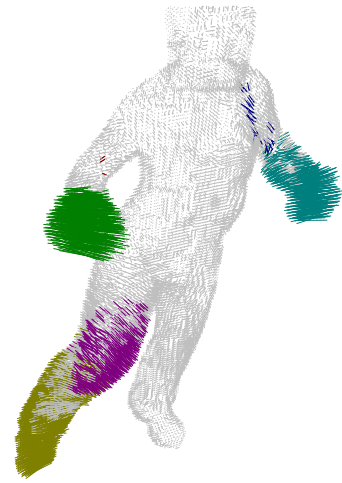


Figure 2. Clustered motion flow lines

Constraints We employed the following five constraints to control the frame-to-frame deformation:

1. **Photometric constraint:** a patch in the mesh model should be placed so that its texture, which is computed by projecting the patch onto a captured image at both frame t and $t + 1$, should be consistent irrespectively of onto which image it is projected.
2. **Silhouette constraint:** when the mesh model is projected onto an image plane, its 2D silhouette should be coincide with the observed object silhouette at frame $t + 1$ on that image plane.
3. **Smoothness constraint:** the 3D mesh should be locally smooth and should not intersect with itself.
4. **Motion flow constraint:** a mesh vertex should drift in the direction of the motion flow of its vicinity (Figure 1).
5. **Inertia constraint:** the motion of a vertex should be temporally smooth and continuous.

Motion Property The heterogeneous deformation change the deformation process of each vertex according to its physical and photometric properties in order to represent object actions as a mixture of warping and rigid motions. We categorize vertices into warping or rigid part by clustering the estimated motion flow of drift and the inertia force (Figure 1 and 2). With this clustering result, we assume that we can categorize the vertices into following two types:

Rigid part (Ca-1) an element of a rigid part of the object and should move together with others in the same part, or

Warping part (Ca-2) a vertex corresponding to a part of object surface under free deformation.

Vertex Identifiability As is well known, we can not expect that all the points on the object surface have prominent texture and can be recovered by stereo method. Hence not all the vertices of the mesh model are *identifiable*, and the photo-consistency constraint, which put a vertex on the real object surface based on texture correlation, will not work at such vertices. So we assume that we can categorize the vertices into two types:

Cb-1 a vertex with prominent texture which should *lead* its neighbors, or

Cb-2 others which should be *led* by its neighbors.

We regard a vertex as identifiable if it has consistent and prominent textures in visible cameras, and label as **Cb-1** (identifiable), and as **Cb-2** if not.

Heterogeneous Deformation Algorithm With these two categorizations, the heterogeneous deformation process is designed as follows:

Step 1. Set the given object shape at frame t as the initial shape of the mesh model.

Step 2. Compute roughly estimated motion flow for the drift and the inertia force.

Step 3. Categorize the vertices based on the motion flow:

Step 3.1. By clustering the estimated motion flow, label the vertex whether **Ca-1**: it is an element of a rigid part, or **Ca-2**: it is not.

Step 3.2. Make the springs of vertices labeled as **Ca-1** stiff.

Step 4. Deform the model iteratively:

Step 4.1. Compute forces working at each vertex respectively.

Step 4.2. For a vertex whose identifiability $I(v)$ exceeds a certain threshold, that is, for a vertex labeled as **Cb-1**, let the force of it diffuse to those of neighbors.

Step 4.3. Move each vertex according to the force.

Step 4.4. Terminate if the vertex motions are small enough. Otherwise go back to 4.1 .

Step 5. Take the final shape of the mesh model as the object shape at frame $t + 1$.

Note that for a vertex of type **Ca-2** \wedge **Cb-2**, a vertex without prominent texture or not a part of a rigid part, its position is interpolated by the smoothness constraint, and a vertex of type **Ca-1** \wedge **Cb-1**, a vertex with prominent texture and a part of a rigid part, deforms so as to lead the rigid part which the vertex belongs to.

4. Deformable Mesh Model for Complex 3D Motion Estimation

In this section, we introduce our new algorithm to estimate complex 3D motion including apparent change of the object shape topology. For global topological change of the object, some researches proposed methods to merge or split surfaces[13][2], but we preserve touched surfaces and do not apply mesh-merging operations since we estimate not shape outline but object motion. To preserve touched surfaces, we need new constraint which make touched surfaces repel each other. This is because the definitions of forces employed in the previous sections based on geodesic distance, and do not consider another vertices which are close in Euclidean distance but apart in geodesic distance. So we introduce how we can find such touched, i.e., collided surfaces efficiently and how we define the repulsive force.

4.1. Efficient Collision Detection for Deformable Mesh Model

Global collision detection is a well-known problem in cloth simulation of computer graphics or another physics based simulation. There are several “short-cuts” to detect a collision between special elements, e.g., spheres or functional surfaces, but collision detection for generic triangle meshes falls back basically to a kind of brute-force algorithm.

In this section, however, we propose a short-cut of collision detection for our deformable mesh model. Let us recall that in our deformation, global collisions occurs only at surfaces in touching. On such a touching surface, we can assume that “visible cameras” C_v for each vertex v to be an empty set \emptyset . Here, “visible cameras” C_v of a vertex v is the set of cameras which can observe v and we have already computed C_v for all vertices to compute another forces based on photo- and silhouette-consistency (Section 3). So we can drastically cull out vertices such that $C_v \neq \emptyset$ from collision detection target. After this efficient culling, we apply brute-force algorithm.

We define our collision detection and repulsive force generation algorithm as follows:

Step 1. For all vertices in the mesh, initialize the repulsive force $F_r(v)$ to 0.

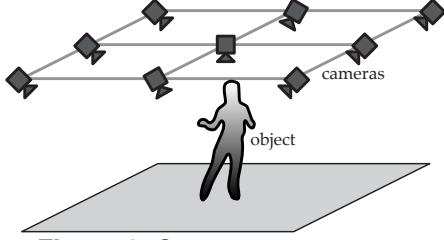


Figure 3. Camera arrangement

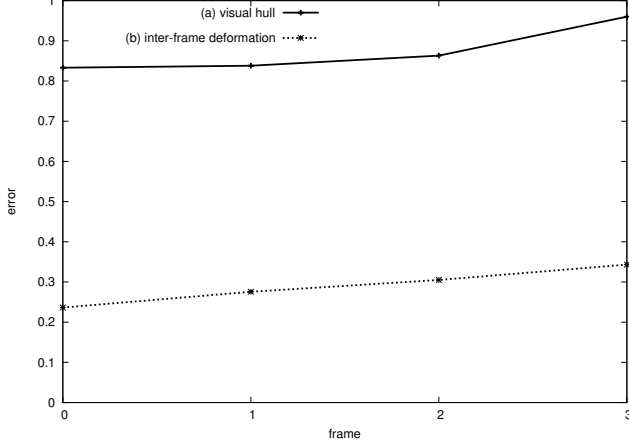


Figure 5. Average shape error

Step 2. Suppose we have a set of vertices such that $V_0 = \{v \mid C_v = \emptyset\}$.

Step 3. For each vertex $v \in V_0$,

Step 3.1. Compute the Euclidean distances to all the others.

Step 3.2. Find vertices such that they are within less than $l_{\min}(v)$ distance, where $l_{\min}(v)$ denotes the minimal length of edges connecting to v . Let $V_d(v)$ denote the set of vertices found for v .

Step 3.3. For each vertex $v' \in V_d(v)$, add following partial repulsive force $f_r(v, v')$ to $F_r(v)$:

$$f_r(v, v') = \frac{\mathbf{q}_{v'} - \mathbf{q}_v}{\|\mathbf{q}_{v'} - \mathbf{q}_v\|^3}, \quad (1)$$

where \mathbf{q}_v denotes the 3D position of the vertex v .

Finally, we add this repulsive force into deformation process described in Section 3.

5. Experiments

5.1. Synthesized Object

Figure 4 shows deformation results of synthesized object using 9 cameras arranged as shown in Figure 3. The left column shows synthesized objects at each frame, the center column shows visual hulls of the object, and the right

Frame	$C_v = \emptyset$	$C_v \neq \emptyset$	Total	Ratio (%)
t	590	7699	8289	92.88
$t + 1$	595	7694	8289	92.82
$t + 2$	595	7694	8289	92.82
$t + 3$	672	7617	8289	91.89

Table 1. Number of vertices such that $C_v = \emptyset$ or not

column shows deformed mesh models. Figure 5 shows average shape error between synthesized object and (a) visual hull, and (b) the results of inter-frame deformation. Here, average shape error is defined as the average distance from each vertex to the nearest point in the synthesized object with height 100. Table 1 shows, from left to right, 1) the frame number, 2) the number of vertices such that no cameras can observe it, i.e., $C_v = \emptyset$, 3) the number of vertices such that $C_v \neq \emptyset$, 4) total count of the mesh vertices, and 5) the percentage of vertices such that $C_v \neq \emptyset$ in the total of mesh vertices, i.e., the percentage of vertices culled out in our collision detection, respectively.

Note that 1) the initial shape of our deformation results are given by an intra-frame shape estimation algorithm, i.e., frame-wise, static 3D shape estimation algorithm[8], 2) it costs about three hours for every frame to deform by PC (Xeon 3.0GHz).

From these results, we can observe that

- The deformable mesh model can cope with global change of topology, i.e., can “touch” itself.
- At “elbow” of each deformation results, we can find folded surfaces in the inter-frame deformation results while the visual hulls do not have such folds. This is because that such region is represented as a visible surface at the first frame, but it is turned to be invisible after some frames.
- Shape errors of our deformation results increase frame by frame, since it cannot avoid error accumulation.
- Our collision detection algorithm can cull out most of vertices based on $C_v = \emptyset$ or not (more than 90%, in Table 1).

5.2. Real Object

Figure 7 illustrates the results of motion estimation of “hand-shaking” two persons. We reconstructed static 3D shape at frame t with conventional frame-wise algorithm[8], and deform it so as to be the shape at $t + 1$ and $t + 2$. In this experiment, we used 15 cameras circumnavigating the object (Figure 6). Figure 8 and 9 illustrate the inter-frame deformation for the object with time-varying global topology. The columns of Figure 8 show, from left to right, the captured images, the visual hulls generated by the

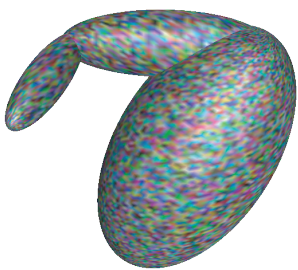
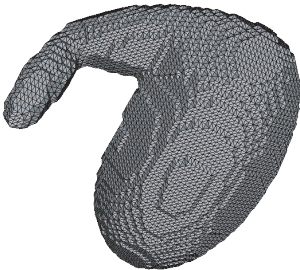
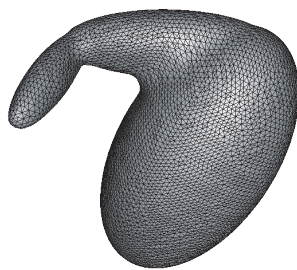
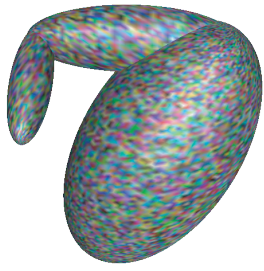
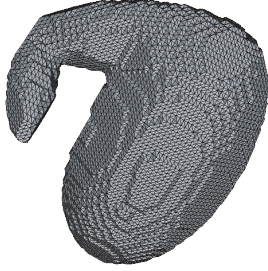
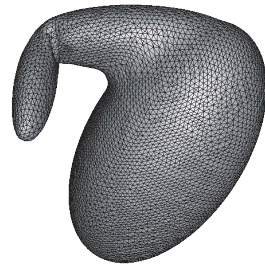
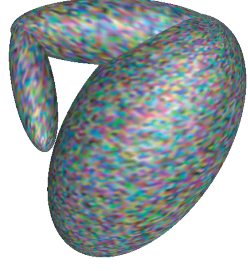
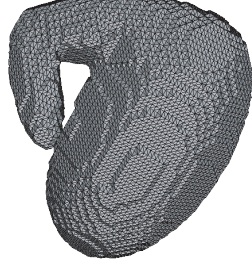
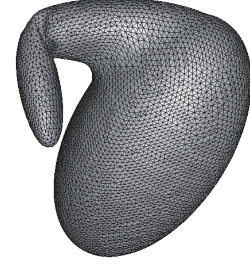
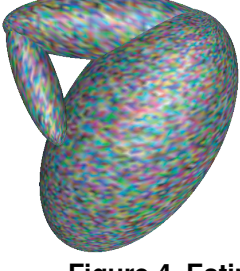

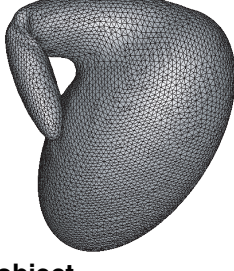
	Synthesized	Visual hull	Inter-frame deformation
t			
$t+1$			
$t+2$			
$t+3$			

Figure 4. Estimation results of synthesized object

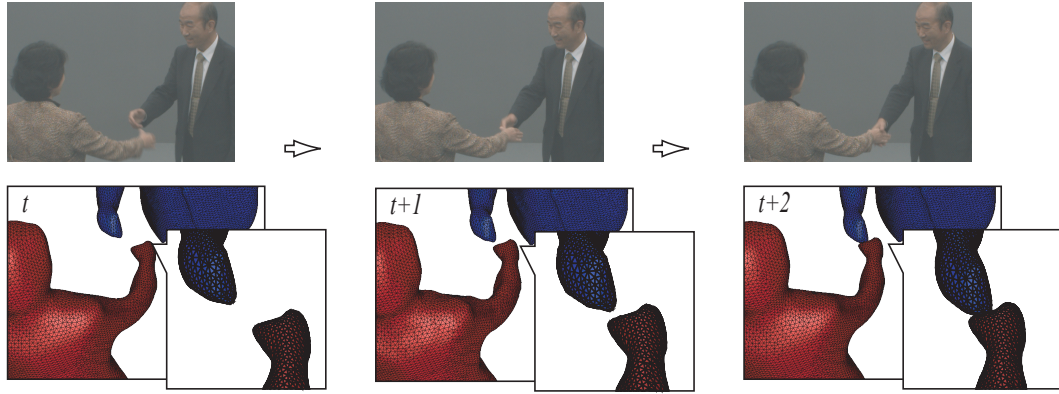


Figure 7. Results of multiple objects estimation

Acknowledgement

This research was supported by Ministry of Education, Culture, Sports, Science and Technology under the Leading Project, “Development of High Fidelity Digitization Software for Large-Scale and Intangible Cultural Assets.”

Figure 6. Camera arrangement for “hand-shaking” sequence

frame-wise discrete marching cubes method, the shape by a conventional frame-wise shape estimation algorithm, and the mesh models deformed by our inter-frame deformation method respectively.

These results show that our multi-object motion estimation can be used to estimate complex, multiple human actions including body touching, i.e., global topological change of the object shape.

6. Conclusion

In this paper, we proposed an algorithm to estimate 3D complex motion of multiple object using deformable mesh model. We introduced repulsive force between vertices based on Euclidean distance and efficient collision detection algorithm to keep hidden and collided mesh surfaces be touched each other.

Our inter-frame deformable mesh model produces dense and non-rigid 3D motion of the objects. It only requires a 3D shape of initial frame given by conventional 3D shape estimation methods, and does not require any object shape model given a priori. Moreover, estimated dense and per-vertex translation vectors can be used as the input data of inter-frame 3D mesh compression algorithms which require topologically consistent meshes[3][6].

References

- [1] A. Bottino and A. Laurentini. A silhouette-based technique for the reconstruction of human movement. *CVIU*, 83:79–95, 2001.
- [2] J. Brendo, T. M. Lehmann, and K. Spitzer. A general discrete contour model in two, three, and four dimensions for topology-adaptive multichannel segmentation. *PAMI*, 25(5):550–563, 2003.
- [3] H. M. Briceño, P. V. Sander, L. McMillan, S. Gortler, and H. Hoppe. Geometry videos: A new representation for 3d animations. In *Proc. of SIGGRAPH*, pages 136–146, 2003.
- [4] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: combining multi-image stereo and shading. *IJCV*, 16(1):35–55, 1995.
- [5] T. Heap and D. Hogg. Towards 3d hand tracking using a deformable model. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, pages 140–145, 1996.
- [6] L. Ibarria and J. Rossignac. Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity. In *Proc. of SIGGRAPH*, pages 126–135, 2003.
- [7] J. Isidoro and S. Sclaroff. Stochastic mesh-based multiview reconstruction. In *Proc. of 3DPVT*, pages 568–577, Padova, Italy, July 2002.
- [8] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation and high fidelity visualization for 3d video. *CVIU*, 96:393–434, Dec. 2004.
- [9] S. Nobuhara and T. Matsuyama. Heterogeneous deformation model for 3d shape and motion recovery from multi-viewpoint images. In *Proc. of 3DPVT*, pages 566–573, Thessaloniki, Greece, September 2004.


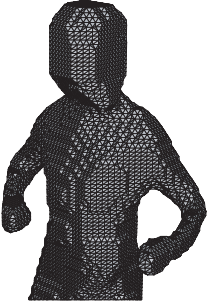
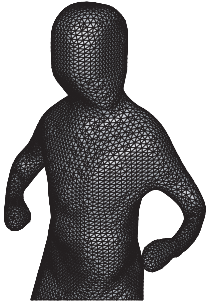
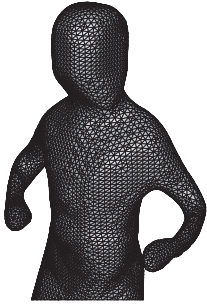

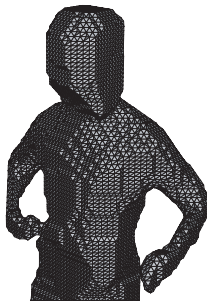
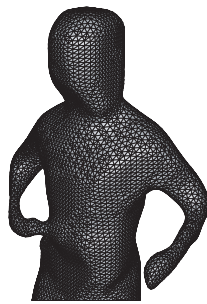
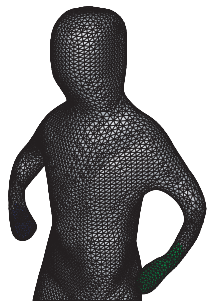

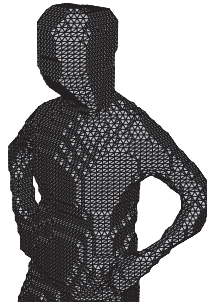
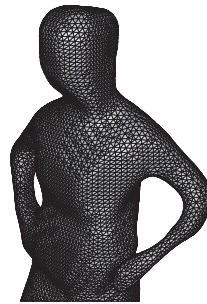
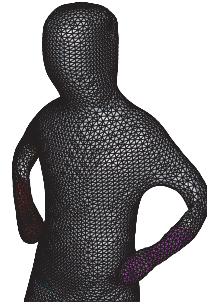
	Observed Image	Visual hull	Intra-frame deformation	Inter-frame deformation
t				
$t+1$				
$t+2$				

Figure 8. Successive deformation results

- [10] R. Plänkers and P. Fua. Articulated soft objects for multiview shape and motion capture. *PAMI*, 25(9):1182–1187, 2003.
- [11] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *Proc. of ICCV*, pages 349–356, Beijing, China, Oct. 2005.
- [12] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6d. In *Proc. of CVPR*, June 2000.
- [13] A. J. Yezzi and S. Soatto. Stereoscopic segmentation. *IJCV*, 53(1):31–43, 2003.

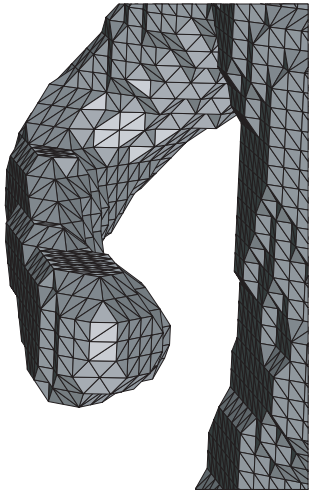
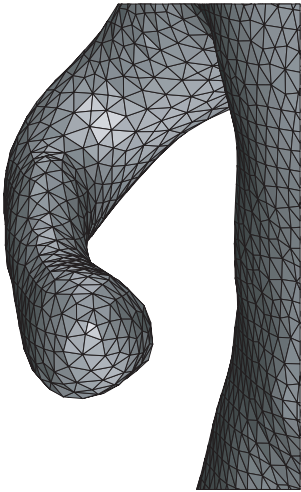
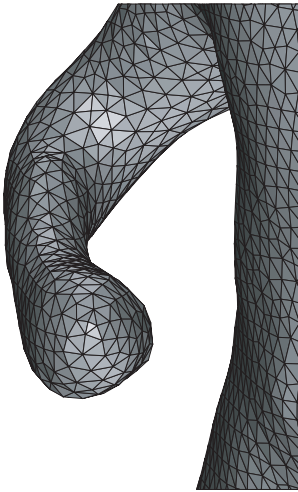
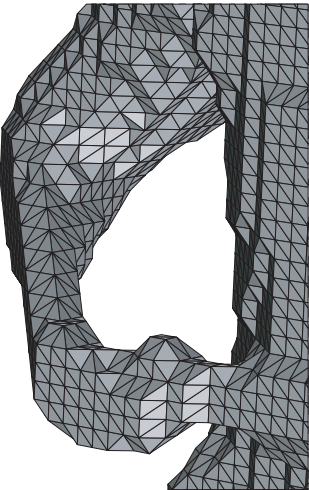
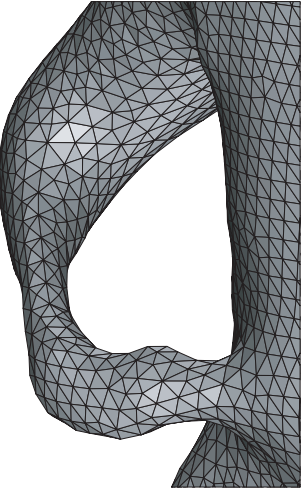
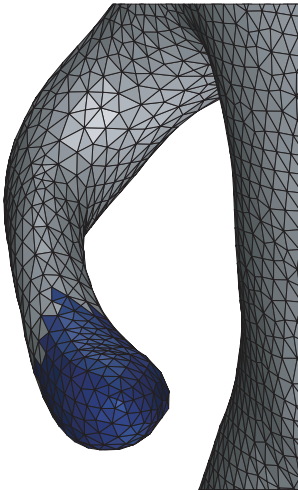
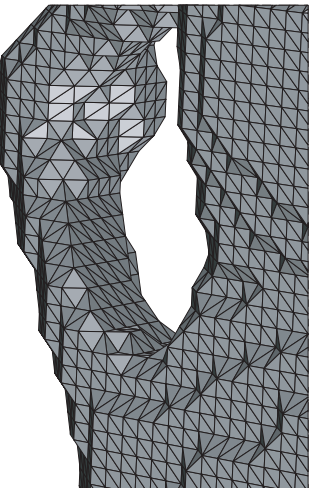
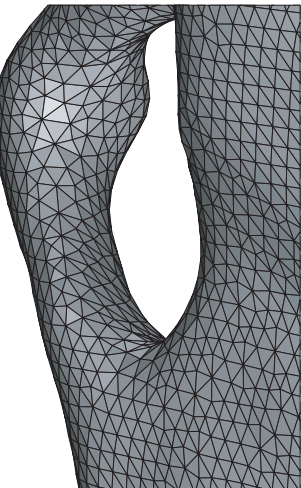
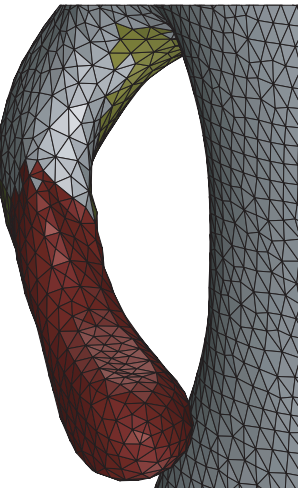
	Visual hull	Intra-frame deformation	Inter-frame deformation
t			
$t+1$			
$t+2$			

Figure 9. Successive deformation results (detailed)