Time-Varying Surface Reconstruction of an Actor's Performance

Ludovic Blache[1](✉), Mathieu Desbrun[2,3], Céline Loscos[1], and Laurent Lucas[1]

[1] University of Reims Champagne-Ardenne, Reims, France
ludovic.blache@univ-reims.fr
[2] Caltech, Pasadena, USA
[3] INRIA, Sophia-Antipolis, France

Abstract. We propose a fully automatic time-varying surface reconstruction of an actor's performance captured from a production stage through omnidirectional video. The resulting mesh and its texture can then directly be edited in post-production. Our method makes no assumption on the costumes or accessories present in the recording. We take as input a raw sequence of volumetric static poses reconstructed from video sequences acquired in a multi-viewpoint chroma-key studio. The first frame is chosen as the reference mesh. An iterative approach is applied throughout the sequence in order to induce a deformation of the reference mesh for all input frames. At first, a pseudo-rigid transformation adjusts the pose to match the input visual hull as closely as possible. Then, local deformation is added to reconstruct fine details. We provide examples of actors' performance inserted into virtual scenes, including dynamic interaction with the environment.

1 Introduction

Multi-view reconstruction of an actor's performance is an innovative, non-invasive technique for computing a 3D avatar of an actor and placing it as an animated character in a virtual scene. It involves a *virtual cloning* system with a set of multi-viewpoint cameras in an indoor studio that generate an animated 3D model of an actor's performance, without the need for the traditional markers typically used in motion capture. From this 3D data, a temporally-coherent surface mesh needs to be constructed to facilitate post-production editing.

Model-based multi-view reconstruction approaches use a template model representing an actor – typically, an articulated mesh of a generic human body. A high-quality template model is often obtained through reconstruction from an actor's 3D scan [1]. Multi-view reconstruction is then achieved by deforming this template in time according to a set of directives (optical flow or silhouette matching) extracted from the multi-viewpoint video inputs. Vlasic *et al.* [2] and Gall *et al.* [3] use a predefined skeleton to match the template model with a set of poses defined by silhouettes or visual hulls, before applying local deformation of the template to match free-form elements such as clothes or hair.

G. Bebis et al. (Eds.): ISVC 2015, Part I, LNCS 9474, pp. 92–101, 2015.
DOI: 10.1007/978-3-319-27857-5_9

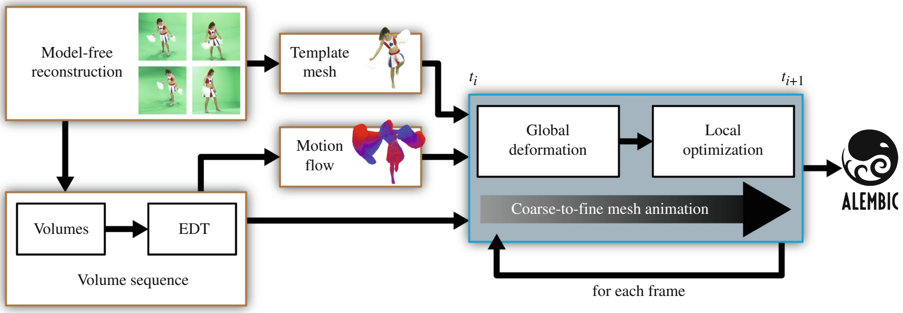

Fig. 1. General pipeline. After a model-free reconstruction, we compute a volume sequence and a template mesh. A motion flow is computed from the volume sequence (Sect. 2.1). The mesh is then animated, according to this motion flow, by a two-step mesh deformation (Sects. 2.2 and 2.3). The animated mesh is finally saved into a file to be used in a post-production software.

However, assuming a specific template model is often too restrictive to capture arbitrary motion sequences: for instance, skeleton-based approaches lead to strong limitations when applied to actors wearing loose costumes (dresses, coats) or accessories (bags, hats) without ad-hoc handling of additional features. This restrictive setup is closer to a markerless motion capture than a reconstruction of the actual scene's content. In addition, TV production stages are often only equipped with video cameras, preventing the use of markers or scans.

In this paper we propose a new approach to reconstruct a time-varying mesh with a fixed connectivity in time from an actor's performance on a typical TV production stage with omnidirectional video capture. Moreover, while many multi-view reconstruction studios use a *model-free* approach that generates a 3D model for each frame of the multi-view video sequence, we facilitate subsequent editing by providing a temporally consistent triangle mesh of the performance through incremental mesh deformation guided by the estimated motion flow.

1.1 Previous Work

Several approaches have been proposed to achieve temporal consistency from this kind of model-free reconstruction, which usually produce a sequence of poses (*i.e.*, a static reconstruction of the scene at each frame of the multi-viewpoint videos). Li *et al.* [4] developed a temporally consistent completion of scanned meshes' sequences, using a deformation graph, to establish pairwise correspondences between consecutive frames. A mesh-tracking method [5–7] can match several meshes according to curvature or texture criteria, from which one can compute the motion flow describing the movements of an actor between two frames [8]. Nobuhara and Matsuyama [9] computes a motion flow from volumetric data, based on a voxel matching algorithm. This voxel-based approach can be made drastically more robust for visual hulls if one considers voxels' orientation

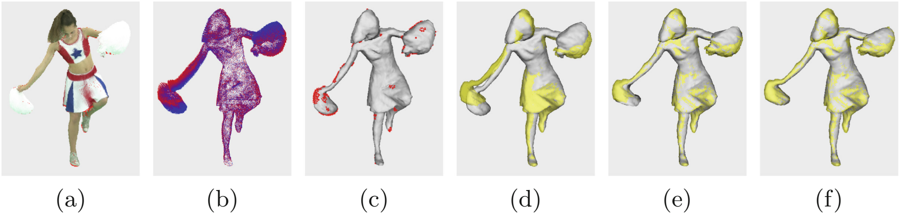

Fig. 2. Our approach at a glance. Starting from two consecutive frames in a sequence of colorized volumes (a), we first compute a motion flow (b); we then sample a set of anchor vertices (c) on the current mesh (in grey). To deform the current mesh towards the next visual hull in yellow (d), we perform a global deformation of the mesh based on the displacements of the anchors (e), before applying a local optimization to finely match the target visual hull (f). This process is repeated throughout the whole sequence to match successive poses (Color figure online).

and texture to help with matching as proposed in Blache *et al.* [10]. Once the motion flow is determined, a template structure needs to be animated. Several types of mesh deformation approaches could be applied in this context. Skeleton-based animation techniques have been shown especially efficient for motion capture. However, it cannot handle complex, non-rigid motions. Recently, 3D surface registration based on energy minimization have been also proven robust for mesh deformation [11]. Our approach is inspired from these different classes of methods, but resolves some of their limitations such as noise sensitivity and lack of genericity.

1.2 Outline

We propose to animate a template mesh derived from the first frame of the sequence of captured volumes to match the subsequent captured frames. Our input data are described in Sect. 2. In order to reconstruct high-frequency movements and to be stable even for less accurate input models, we combine motion flow to ARAP surface modeling, adapted with automatic anchors' selection (Sect. 2.1). The global deformation based on ARAP matching [12] will ensure robustness of the mesh deformation even in the presence of noisy motion flow (Sect. 2.2). Local adjustments will improve quality both of the mesh and of the match between the mesh and the volumetric input data (Sect. 2.3). This deformation process is iterated between pairs of frames until the end of the sequence to consider. Our approach is markerless, fully automatic through the entire pipeline, and is robust to a variety of scenes involving actors wearing costumes and accessories. The different steps are illustrated in Fig. 2.

2 Method Overview

Our input is a sequence of digital volumes obtained by a *visual hull* reconstruction from a multi-view video sequence [13]. Each volume of this sequence represents the actor at time t_i, so that the set of n volumes represents the motion of the actor from t_0 to t_{n-1}. The reconstructed volumes are binary digital volumes with each voxel tagged as 0 for empty or 1 for covering or straddling the actor's spatial occupancy. Straddling (surface) voxels are also assigned an RGB color based on the video inputs.

We begin by computing a Euclidean distance transform (*EDT* [14]). Each voxel of the volume is thus assigned a positive value which corresponds to the Euclidean distance to the closest boundary of the actor, so that this volumetric description can be seen as a grey-level 3D picture. We construct an initial (template-like) mesh based on the first volume of the sequence by extracting its zero levelset using a *marching cubes* algorithm. Laplacian smoothing and mesh simplification are then applied to ensure that each resulting triangle is non-degenerate, thus avoiding numerical artifacts in our subsequent tracking.

2.1 Motion Flow and Anchors' Selection

In order to compute a motion flow from our set of volumetric images, we use the method described in [10] to compute voxel matching based on both local geometry and color between consecutive poses. The result is a set of motion vectors for each surface voxel throughout the volume sequence. The matching is performed with a distance function computed between two surface voxels from two successive frames. This distance is computed according to several criteria: normal orientation, color and Euclidean distance. We use this matching score between two voxels as a degree of confidence associated to each vector of the motion flow, noted w_a. We select a set of vertices at time t_i to be *anchor* points. As these anchors will drive the global deformation of the character, we select the vectors associated with the largest displacements and highest confidence. Mesh vertices at t_i are thus ordered according to their corresponding $EDT_{i+1} + w$ values. We then select a fixed percentage (10 % of the total number of vertices in our implementation) of the highest ranked vertices. A few anchors in static regions (1 %) are also randomly added to guarantee that the immobile parts of the actor's body will not be deformed. This subsampling of the motion flow allows robustness since a global deformation is derived from two subsequent 3D volumes by removing the high frequency noise that typically impairs proper tracking. The scores w_a will still be used as weights to adjust how strongly we enforce the matching of these anchors in the global deformation step for additional robustness. Note that while this anchors' sampling method is particularly suited to our context, it could easily be adapted to other model-based approaches.

2.2 Pseudo-Rigid Mesh Animation

The mesh at t_i now needs to be advected in the motion flow based on the displacement of the reduced set of sampled anchors. We use a variational approach to our global mesh deformation by searching for a As-Rigid-As-Possible (ARAP [12]) deformed mesh M' with locally rigid transformations, while retaining the final positions of anchor points as much as possible (Fig. 2e).

Formulation. We minimize the following energy:

$$E(M') = E_{ARAP}(M') + E_{ANC}(M'),$$

where E_{ARAP} is the *As-Rigid-As-Possible* energy:

$$E_{ARAP}(M') = \sum_{i=1}^{n} \sum_{j \in N(i)} \gamma_{ij} \left\| (p'_i - p'_j) - R_i(p_i - p_j) \right\|^2 , \tag{1}$$

with $N(i)$ denoting the one-ring neighborhood of i. The terms p_i and p'_i represent the 3D positions of the vertex i, before and after applying the local transformation R_i. Note that if p_i is an anchor, the position p'_i is initialized by applying the associated motion vector to the initial position of the vertex. The weight γ_{ij}, associated with the edge between p_i and p_j, can be computed according to the cotangent weight method, or simply set to 1. Moreover, E_{ANC} is a quadratic energy measuring the error in the displacement of the n_a anchors:

$$E_{ANC}(M') = \sum_{i=1}^{n_a} w_{a_i} \left\| p'_i - p_i \right\|^2 , \tag{2}$$

where the weight w_{a_i} of the anchors represents the degree of confidence given to an anchor point, as described in Sect. 2.1.

Solver. The optimality condition for the minimum of our energy basically mirrors the result of [12], to which terms coming from the quadratic form (Eq. 2) are added. That is, the optimal positions p' must satisfy:

$$\sum_{j \in N(i)} \gamma_{ij}(p'_i - p'_j) + w_{a_i} p'_i = \sum_{j \in N(i)} \frac{\gamma_{ij}}{2}(R_i - R_j)(p_i - p_j) + w_{a_i} p_i \tag{3}$$

where R_i is a local rotation best matching p_i and its one ring to p'_i. The global deformation is thus computed by iteratively solving a linear system and an optimal set of rotations matrices: we begin by computing the set of $\{p'_i\}_i$ that satisfy the optimal condition for a fixed set of initial rotations $\{R_i\}_i$ by solving a linear system of the form:

$$Lp' = b$$

where L corresponds to the Laplacian operator applied to the mesh M' in which we add the w_{a_i} weights related to each anchor point (Eq. 2) on the diagonal, and b is a column matrix which contains the righthand side of Eq. 3. Optimal rotations R_i are computed through singular value decomposition (SVD) from the positions of p_i and p'_i as derived in [12]. These two steps are repeated until convergence.

2.3 Local Optimization

After the global deformation step, details of the pose due to non-rigid deformation (such as cloth folds or hair) can still be missing. Mesh quality may also degrade over time as large deformation occurs. Local optimization and regularization are thus still necessary for the mesh to better adjust to the new pose's silhouette. We therefore

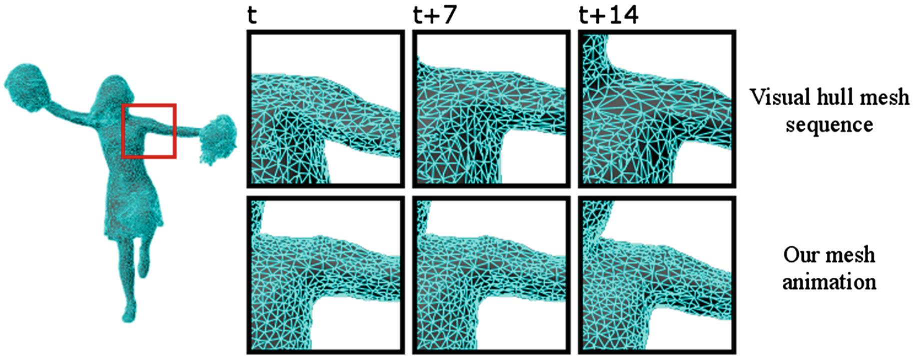

Fig. 3. Comparison between the temporally inconsistent mesh sequence from a model-free reconstruction (top) and our animated mesh (bottom).

compute local vertex displacements based on both fitting accuracy and regularization as follows.

Regularization. We regularize the mesh by applying a spring force per vertex to favor equi-length edges:

$$f_r(p_i) = \alpha \sum_{j \in N(i)} (\|p_i - p_j\| - \bar{r}_i) \frac{p_i - p_j}{\|p_i - p_j\|}$$

where α is a fixed stiffness coefficient and p_j is a vertex from the one-ring neighborhood of p_i, while the rest length $\bar{r}_i$ is set to the current average length of the edges adjacent to p_i. We prevent shrinking of the shape by using only the tangential component of the resulting vector.

Silhouette fitting. Using the EDT zero isovalues (Sect. 2.1), we also locally inflate or deflate the mesh towards the visual hull surface by adding a "balloon" force expressed as:

$$f_s(p_i) = \sum_{j \in N(i)} EDT(p_j) \boldsymbol{n}_{\boldsymbol{p}_j}$$

with $\boldsymbol{n}_{\boldsymbol{p}_i}$ and $EDT(p_i)$ being the normal vector and the EDT value at p_i, respectively. Only the normal component of this force term is used.

Integration. The resulting vectors f_r and f_s are added to obtain a displacement in time for each vertex. This displacement is integrated over a fixed 200 time steps between pose t_k and t_{k+1} by updating position and velocity of each vertex (assumed to be all of unit mass) using a simple Runge-Kutta explicit integrator.

3 Results and Discussion

We tested our method on several datasets obtained through volumetric visual hull reconstruction. The *cheerleader* sequence contains 200 frames with an average $180 \times 270 \times 170$ voxel resolution and a 19234-vertex template. The *astronaut* sequences contains 25 volumes, with an average $150 \times 120 \times 330$ resolution and a 8048-vertex mesh.

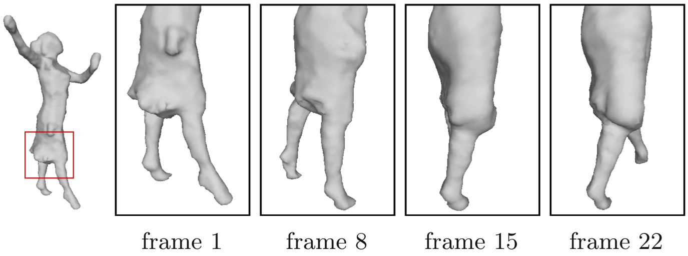

Fig. 4. Example of free-moving clothes' tracking in the *dancer* sequence.

These two datasets come from an indoor studio shoot using a 24-camera rig. The *dancer* sequence was generated using the multi-viewpoint images provided by the GrImage platform[1] with an average 150 × 100 × 300 voxel resolution. Due to the low number of cameras (8 viewpoints) and their low resolution, this dataset produces coarse visual hulls. We also tested our method on the *capoeira* sequence, using the multi-view video inputs described in [1][2] with a resolution of $200 \times 275 \times 200$. All timings were achieved on a 64 bit Intel Core i7 CPU 2.20 GHz. Results from these sequences are presented in Fig. 6, demonstrating robustness of our approach in light of the coarseness of the input volumes.

The global deformation (Sect. 2.2) needs at most 200 iterations to converge. The local mesh optimization (Sect. 2.3) is applied with equal weights for the two forces f_r and f_s. A maximum of 200 iterations is necessary for the numerical integration. The total mesh processing is performed in an average of 110 seconds for each frame of the *cheerleader* sequence, and 80 seconds for *astronaut* and *dancer*. Our mesh animation approach described in Sect. 2.2 leads to a locally rigid deformation, which preserves the mesh structure during the whole sequence. It should also be noted that our use of weights based on the reliability of the anchors nicely extends the ARAP modeling technique, rendering it particularly robust to the inherent noise present in the motion flow. This improvement does not require higher computational costs since the added anchor energy we proposed (Eq. 2) only adds diagonal elements in the Laplacian-like matrix involved in the original ARAP method. The final local optimization step (Sect. 2.3) then adapts the mesh to the non-rigid part of the motion, allowing the recovery of detail in clothes and accessories after the global deformation has been properly recovered. The *cheerleader* dateset shows that shape of the pom-poms is correctly adjusted after the global deformation phase (Fig. 2e and f). With the *dancer* sequence, we show that the mesh correctly tracks the shape of the moving dress (Fig. 4). Our mesh animation method leads to an adaptation of the template during the sequence, avoiding some of the model-based inconveniences (as in, e.g., [1]) where the tracked model retains surface details (clothing folds) from the initial pose during the whole sequence. With the *capoeira* sequence, the lower quality of the multi-view images and silhouettes produces a noisy and damaged reconstruction. The visual hulls of this sequence contains many irregularities such as occlusion artifacts and holes in the character's shape. Yet, our template deformation matches the silhouettes better (Fig. 5) than [1], which offers a high quality reconstruction but mostly misses the clothes' deformation. Quantitatively,

[1] http://4drepository.inrialpes.fr/.

[2] http://resources.mpi-inf.mpg.de/siggraph08/perfcap/.

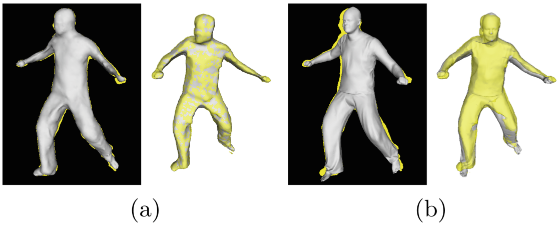
(a) (b)

Fig. 5. Capoeira sequence. Our method (a), despite the low quality of the visual hull, matches the silhouette better than the model-based method proposed in [1] (b). Left: silhouette overlap. Right: Comparison between visual hull (yellow) and 3D mesh (grey) (Color figure online).

the average Hausdorff distance computed between the visual hull of the target pose and the animated template resulting from the model-based method [1] is 0.011933, while our method obtains an average of 0.003291.

Our temporally coherent mesh is perfect for postproduction editing as it can be easily placed in a virtual environment as a simple animated character, directly exported through, e.g., the Alembic file format. A virtual camera (with an arbitrary trajectory) can then be used without being limited by the characteristics of the shooting studio. The rendering of the virtual scene is noticeably easier with this animated object than with mesh sequences when each pose of the sequence needs to be loaded before the rendering of the corresponding frame. Generating a mesh for which only mesh vertices evolve in time (Fig. 3) has multiple additional advantages. First, it noticeably reduces the flickering effect of visual hull reconstruction. Second, vertices can be used to anchor virtual accessories (e.g. virtual makeup). Third, collision with virtual objects (clothes or other) and environment is easy to detect and handle as one can rely on temporal coherence of the vertices. At last, the animated mesh can keep the same texture during the animation, instead of computing a new one for each frame. For long sequences,

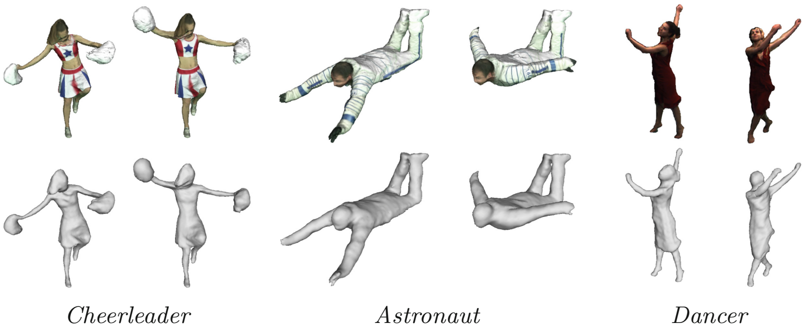
Cheerleader *Astronaut* *Dancer*

Fig. 6. Results of our approach for the *Cheerleader*, *Astronaut* and *Dancer* sequences. Comparison between the original visual hull reconstruction (top) and the temporally consistent mesh (bottom).

it is preferable to keep a single UV map for the whole sequence, associated with an animated texture to prevent a visual texture *sliding* effect.

In several extreme cases, our local regularization step degrades the details of very thin features as they are of a size too close to the size of a grid element. A local optimization with subgrid accuracy could solve this issue, most likely at the cost of a significantly increased computational complexity. Our approach also assumes that the initial topology is kept throughout the sequence. However, changes in the topology of the visual hull could occur in the captured sequence, possibly due to occultations if not enough view angles are available. Currently, these events are not supported by our system and requires user interaction to correct. In the future, stereo-matching could be used to improve the accuracy and quality of the volume sequences. Alternatively, one could also handle topology changes through, for instance, the method proposed by Bojsen-Hansen *et al.* [15]: they proposed a surface-tracking based on a non-rigid registration and address the issue of topology changes by partially resampling the mesh.

4 Conclusion

In this paper, we have proposed a new approach for generating a time-evolving triangle mesh representation from a sequence of binary volumetric data representing an arbitrary, possibly complex and unstructured motion of an actor with arbitrary costumes and accessories. We first compute a motion flow of the sequence from a voxel-matching algorithm. Using the visual hull as a prior, we then animate a template mesh, generated by a surface reconstruction of the first volume, via as-rigid-as-possible, detail-preserving transformations guided by the motion flow and based on a sparse set of weighted anchors. A final local optimization adjusts the mesh to better match the mesh shape to the current visual hull, leading to a robust, temporally-consistent mesh reconstruction of the motion.

Acknowledgments. M. Desbrun is partially funded by the National Science Foundation (CCF-1011944 grant), and gratefully acknowledges being hosted by the TITANE team in the context of an INRIA International Chair. We would like to thank our partner XD Productions. This work has been carried out thanks to the support of the RECOVER3D project, funded by the *Investissements d'Avenir* program. Some of the captured performance data were provided courtesy of the Max-Planck-Center for Visual Computing and Communication (MPI Informatik/Stanford) and Morpheo research team of INRIA.

References

1. de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. ACM Trans. Graph. **27**, 98:1–98:10 (2008)
2. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. ACM Trans. Graph. **27**, 97:1–97:09 (2008)
3. Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1746–1753 (2009)

4. Li, H., Luo, L., Vlasic, D., Peers, P., Popović, J., Pauly, M., Rusinkiewicz, S.: Temporally coherent completion of dynamic shapes. ACM Trans. Graph. **31**, 2:1–2:11 (2012)
5. Starck, J., Hilton, A.: Correspondence labelling for wide-timeframe free-form surface matching. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1–8 (2007)
6. Varanasi, K., Zaharescu, A., Boyer, E., Horaud, R.: Temporal surface tracking using mesh evolution. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 30–43. Springer, Heidelberg (2008)
7. Tung, T., Matsuyama, T.: Dynamic surface matching by geodesic mapping for 3D animation transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1402–1409 (2010)
8. Petit, B., Letouzey, A., Boyer, E., Franco, J.S.: Surface flow from visual cues. In: International Workshop on Vision, Modeling and Visualization (VMV), pp. 1–8 (2011)
9. Nobuhara, S., Matsuyama, T.: Heterogeneous deformation model for 3D shape and motion recovery from multi-viewpoint images. In: Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 566–573 (2004)
10. Blache, L., Loscos, C., Nocent, O., Lucas, L.: 3d volume matching for mesh animation of moving actors. In: EG Workshop on 3D Object Retrieval, pp. 69–76 (2014)
11. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Dynamic 2D/3D registration. In: Holzschuch, N., Myszkowski, K. (eds.) Eurographics Tutorial. The Eurographics Association (2014)
12. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Proceedings of Symposium on Geometry Processing, pp. 109–116 (2007)
13. Laurentini, A.: Visual hull concept for silhouette-based image understanding. IEEE Trans. Pattern Anal. Mach. Intell. **16**, 150–162 (1994)
14. Saito, T., Toriwaki, J.I.: New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. Pattern Recogn. **27**, 1551–1565 (1994)
15. Bojsen-Hansen, M., Li, H., Wojtan, C.: Tracking surfaces with evolving topology. ACM Trans. Graph. **31**, 53:1–53:10 (2012)