Iterative Deformable Surface Tracking in Multi-View Setups

Cedric Cagniart, Edmond Boyer, Slobodan Ilic

Iterative Deformable Surface Tracking in Multi-View Setups

Cedric Cagniart
CAMP
TU Munich
cagniart@in.tum.de

Edmond Boyer
INRIA Alpes
Grenoble
edmond.boyer@inrialpes.fr

Slobodan Ilic
CAMP
TU Munich
slobodan.ilic@in.tum.de

Abstract

In this paper we present a method to iteratively capture the dynamic evolution of a surface from a set of point clouds independently acquired from multi-view videos. This is done without prior knowledge on the observed shape and simply deforms the first reconstructed mesh across the sequence to fit these point clouds while preserving the local rigidity with respect to this reference pose. The deformation of this mesh is guided by control points that are randomly seeded on the surface, and around which rigid motions are locally averaged. These rigid motions are computed by iteratively re-establishing point-to-point associations between the deformed mesh and the target data in a way inspired by ICP. Our method introduces a way to account for the point dynamics when establishing these correspondences, a higher level rigidity model between the control points and a coarse-to-fine strategy that allows to fit the temporally inconsistent data more precisely. Experimental results, including quantitative analysis, on standard and challenging datasets obtained from real video sequences show the robustness and the precision of the proposed scheme.

1. Introduction

Multi-camera setups allow to recover shapes from the observed scenes by exploiting the spatial redundancy in the observed images. However, even when dealing with temporal sequences, most methods concentrate on providing a visually convincing reproduction of the shapes by treating each time frame as a separate static multi-view 3D reconstruction problem. Many algorithms are quite successful at solving this problem [14] and allow to build precise photometric models. However, independent shape models are just a fraction of the information one might want to extract from multiple videos. For example, capturing body motion or editing geometry in a temporally consistent way are two applications which require to also capture the motion and deformation of the reconstructed surfaces. This requirement has spawned a number of works targeted at exploiting the temporal redundancy in the acquired data to provide a dense surface tracking over time.

Figure 1. Results on the Flashkick sequence. The top row shows a wireframe versions of the deformed mesh we recover overlaid on the original image from the second camera. The bottom row shows shaded models of the same mesh.

Many of these works tackle surface tracking by establishing cross parametrizations between shapes independently reconstructed at adjacent time frames. They usually do so by first matching sparse visual and geometric features and then propagating this information over the surface using smoothness constraints. The strength of these methods is that the motion can be estimated without any assumption on the observed geometry. However, these approaches, being recursive by nature, make it difficult to prevent the incremental build-up of tracking errors. Model-based strategies perform better at providing long term accurate tracking. Among them, the methods that deform a reference surface over time are of particular interest as their output is by nature temporally consistent. Although deforming a reference surface limits the range of possible deformations and prevents the handling of topology changes, recent works [8, 20, 10] have demonstrated the efficiency of this strategy.

The algorithm we introduce in this paper uses a photo-

consistent mesh, e.g reconstructed from the first frame, as a reference model and deforms it across time to fit independently reconstructed sets of points and normals. The deformation framework we use was presented in [15] for interactive modelling purposes and relies on a number of constrained vertex positions to infer a global deformation preserving local differential coordinates. In contrast to interactive modelling, where those constraints are manually given and are noise free, we compute them automatically from challenging data that exhibits fast motion, large deformations, and topology changes. Our method is inspired by the Iterative Closest Point algorithm as it iteratively re-estimates point associations between the deformed mesh and the target data, and averages rigid motions locally around control points randomly seeded on the surface. The displacements of these control points are used as constraints in the mesh deformation framework. The nature of our input data makes achieving robust and reliable performance challenging, and has motivated the ideas presented in this paper. First, we account for the temporal nature of the problem by penalizing dynamically unlikely point associations. Second, we introduce a way of diffusing information between control points to limit the occurrence of degenerate mesh configurations. Third, we propose a coarse-to-fine approach where an increasing number of control points is progressively reseeded to recover higher frequency geometric details. The whole approach is purely geometric and stays temporally local as it reseeds control points independently at each time frame.

In the remainder of this paper we overview related works, detail our contribution, and present the associated results before concluding.

2. Related Work

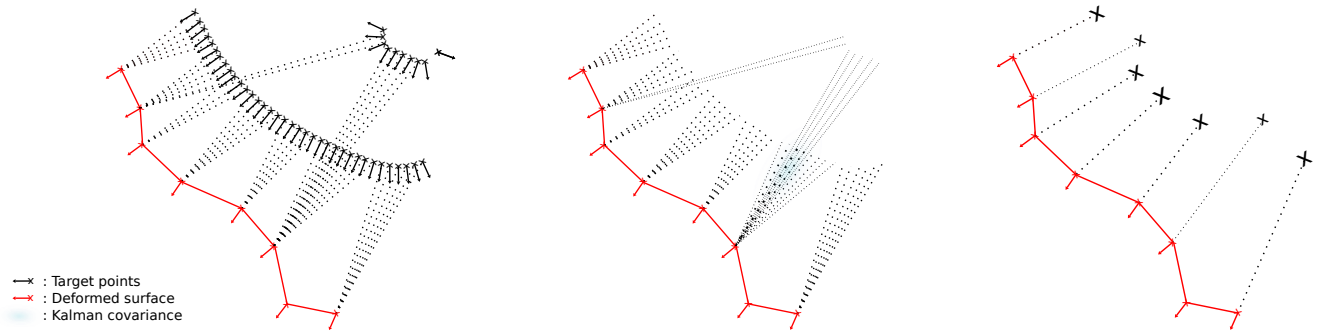
One of the major challenges when capturing the deformation of a surface from multiple videos resides in the sparsity and the non uniformity of the distribution of visual features on the surface. Starck and Hilton [17, 16] add edge information to complement corner detectors. Bradley [6] adresses the similar problem of marker-less garment capture by using the boundaries of the garment as anchors to guide the establishment of a consistent mapping between independently reconstructed surfaces. The use of some geometric features has also been explored. Starck and Hilton [16] use the uniformly distributed geodesic-intensity histogram, then regularize the possible assignments using a MRF on the graph of the mesh. Varanasi [19] matches mesh extremities identified as the extrema of the geodesic integral [11], then regularizes the deformation by using the preservation of local differential coordinates and adds a morphing step to handle topology changes. However, as all these methods diffuse the information from these sparse features to the rest of the surface, they need to assess the reliability of these sparse matches to avoid wrong associations. In particular, geodesic features require special care as soon as topology changes occur.

Introducing a strong model reduces the sensitivity to potential wrong sparse matches. A number of algorithms [5, 8, 20] use a reference surface as model. Naveed [1] propagates the sparse features information over the mesh using level sets of harmonic functions. The works by Aguiar, Vlasic and Gall [8, 20, 10], closely related to our approach, deform a template surface while enforcing local rigidity by preserving Laplacian coordinates. In [8], a coarse volumetric mesh is first deformed using visual features and silhouette constraints, then higher frequency deformations are estimated locally using depth maps computed from stereo. In [20, 10] a skeleton model is first fitted in the visual hull, then the corresponding shape estimation is refined by inflating the surface in order to match the silhouettes. These works interestingly show that sparse visual features are not imperatively required to recover a convincing approximation of the surface deformation. However both skeleton and volume preservation are very restrictive assumptions that limit the extensibility of these approaches to more complex scenes.

The presented method relies on a much weaker, purely surface based model. It uses purely geometric information as input data: clouds of points and normals. It is inspired by the Iterative Closest Point algorithm[7, 4] that was initially proposed to register rigid motions of solid objects. Although many algorithms have build on these ideas to address the non-rigid case [9, 2, 3], to the best of our knowledge none applies to large deformations as observed when capturing body motions. The method by Stoll [18] is the closest to ours but does not deal with temporal sequences and uses manually defined constraints to initialise the deformation, which drastically limits outliers in point associations. In contrast, most of the ideas we present in this paper are motivated by the need for robustness with respect to these wrong point assignments: we penalize dynamically unlikely associations, average rigid motions locally and introduce a higher level rigidity model between control points. The coarse-to-fine procedure also plays a role in limiting the sensitivity to wrong associations but is mostly motivated by the need for precision when fitting the target data. These contributions are detailed in the following section.

3. Method

The presented approach belongs to the *tracking by deformation* class of methods which deform the mesh discretization of a reference surface over time. This reference mesh is defined as a graph (ν, τ) and a position function $X^0 : \nu \mapsto \mathbb{R}^3$, where ν is the set of vertices and τ the set of triangles. As our method does not modify the connectivity, the problem limits to re-evaluating the position function



(a) Each vertex from the target point cloud finds its closest compatible vertex. Erroneous points in the point cloud create bad associations.

(b) These point associations are weighted according to their dynamic residual. The erroneous associations end up having small weights.

(c) The positions are averaged with a weighted sum, yielding the proposal position function X' and the weight function W.

Figure 2. Computing the proposal position function on the mesh

across time.

Our algorithm has similarities with the ICP algorithm: it iteratively re-evaluates point to point correspondence between a target point cloud and a mesh which is deformed to minimize the distances between associated points. These two steps of ICP can be seen as follows: first a *proposal position function* is computed from the associations between the deformed mesh and the data. Then this function is regularized by forcing the deformation field on the object to be a rigid motion. In this paper, both of these steps are redefined to the case of large dynamic deformations. In subsection 3.1, we describe the computation of the proposal position function and how to account for the point dynamics over the time. The regularization scheme is introduced in subsection 3.2.

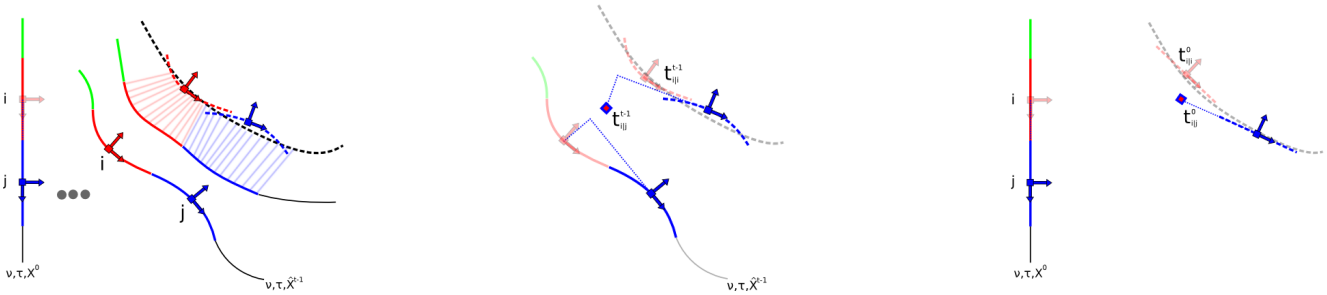
The proposal position function is regularised by forcing the deformation field to be locally as close as possible to a rigid motion. The way locality is defined on the mesh is central in that respect. Control points are seeded on the surface. Around each of them, a rigid motion is averaged on a neighbourhood of fixed geodesic radius. Each of these rigid motions is used to predict a new position for associated control point but also for its neighbours. By computing a weighted average of all these predictions, a target position is obtained and fed to a mesh deformation framework. These two steps are iterated until the control points stop moving. In subsection 3.3 a coarse to fine approach is presented: few control points are first defined to align the deformed mesh with the target points, then more are re-seeded to capture higher frequency details in the target data.

3.1. Computing the proposal position function

The goal in this part of the method is to compute a *proposal position function* $X' : \nu \mapsto \mathbb{R}^3$ and the associated *weight function* $W : \nu \mapsto \mathbb{R}$ which will be used to assess the quality of the information in the rest of the algorithm. In that respect, our method is different from the original ICP algorithm as we adopt the *proposal position function* point of view instead of limiting us to a more restricted point-to-point mapping. Moreover, instead of finding for each vertex of the deformed template the closest point in the target point cloud, we proceed the other way around: we iterate through the target point cloud and have each of its vertex contribute its position to the closest point in the current approximation of the deformed mesh. In the common situation where the tracked surface self intersects and produces a topology change, this means that the unmatched parts of the deformed reference surface simply do not get any contribution and remain with a zero weight function, thus not generating erroneous constraints in the rest of the computation. We also found this approach to be better at getting out of local minima as every point from the target point cloud has an influence on the deformation, and not only the points closest to the current approximation of the deformed mesh.

Similarly to many works which have extended on the original ICP algorithm, the presented method does not establish point to point correspondence by looking for the closest point only, but instead looks for the closest *compatible* point. Both the compatibility and the weighting function can benefit from different types of data input. In their simplest form they will just use the Euclidean distance between the points. Our method defines a compatibility function already used in some ICP variants and prevents associations between points whose normals form an angle superior to $45°$.

We account for *point dynamics* by keeping a Kalman Filter for each vertex of the deformed mesh. We can use its state covariance matrix to define a Mahalanobis distance to the predicted point position and compute a weight for each association. This limits the influence of dynamically unlikely pairings and is done at a very minimal cost as the filters are only updated at the end of computations for each



(a) From left to right: reference pose, result for the previous frame, current approximation, target surface

(b) prediction of c_i and N_i knowing $\mathbf{RT}_j^{t-1}$

(c) prediction of c_i and N_i knowing $\mathbf{RT}_j^0$

Figure 3. Diffusing the rigid motions between neighbouring control points

time frame.

3.2. Regularizing the proposal function

The problem here is to iteratively redefine control points to guide the mesh deformation. In the following paragraphs we detail the mesh deformation framework we use, the way of seeding the control points, and how rigid motions are averaged and diffused around them.

Our regularization scheme relies on "as rigid as possible" deformation using Laplacian mesh processing as presented in [15]. The original paper deals with constraints manually defined by the user on control points, from which the deformation is diffused to the rest of the mesh by trying to preserve local differential coordinates. The Laplacian matrix is computed from the reference mesh using cotangent weights[13] . The coordinates of the mesh we are looking for are stored in the $|\nu| \times 1$ vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$. The target positions for the control points $\mathbf{c}_i$, or constraints, have their coordinates stored in the $|\nu_c| \times 1$ vectors $\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ and their associated weights in the diagonal matrix $\mathbf{W}_c$ (weight 0 if the point is unconstrained). The problem can be written for each coordinate (here for $\mathbf{x}$) as a least-squares system:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{L}\mathbf{x} - \delta\|^2 + \|\mathbf{W}_c(\mathbf{x} - \mathbf{x}_c)\|^2 \qquad (1)$$

where the δ term is computed from the original values of the Laplacian $\mathbf{L}\mathbf{x}^0, \mathbf{L}\mathbf{y}^0, \mathbf{L}\mathbf{z}^0$ modified to account for local rotations of the surface with respect to the reference pose.

Seeding control points As stated in the beginning of this section, our method averages rigid motions on the neighbourhoods of control points seeded on the surface. These control points $\mathbf{c}_i$ are created on the surface along with neighbourhoods N_i of a maximal geodesic radius until the whole surface is covered. To preserve the temporal locality of the approach the seeding process takes place independently at each time frame. In other words we do not keep the same control points all along the time sequence because this would amount to making it part of the model.

Diffusing rigid motions On the neighbourhood N_i around control point $\mathbf{c}_i$, the information $\{(X'(v), W(v)) | v \in N_i\}$ allows to compute an average rigid motions $\mathbf{RT}_i^0$ between the reference pose X^0 and X' using the Horn algorithm[12]. From this rigid motion we predict a target position of the control point $\mathbf{t}_{i|i}^0 = \mathbf{RT}_i^0(\mathbf{c}_i^0)$. Neighbouring control points $\mathbf{c}_j$ also predict target positions $\mathbf{t}_{i|j}^0 = \mathbf{RT}_j^0(\mathbf{c}_i^0)$ for $\mathbf{c}_i$ using their own averaged rigid motions. As shown in Figure 3 we also average rigid motions $\mathbf{RT}_i^{t-1}$ between the result of the previous frame X^{t-1} and X' and obtain the associated $\mathbf{t}_{i|j}^{t-1} = \mathbf{RT}_j^{t-1}(\mathbf{c}_i^{t-1})$. For each of these predictions a residual error $e_{i|j}$ is computed as the average squared distance between the points of N_i transformed by $\mathbf{RT}_j$ and the target point cloud. In a way similar to particle based Bayesian techniques, we use these errors to weight all these predictions and get for each control point $\mathbf{c}_i$ a set of $(\mathbf{t}_{i|j}^0, w_{i|j})$ and $(\mathbf{t}_{i|j}^{t-1}, w_{i|j})$, where the weights are computed according to Equation 2.

$$w_{i|j} = e^{-\frac{e_{i|j}}{l^2}} \qquad (2)$$

where l is the mean edge length of the reference mesh in its original pose.

This procedure roughly amounts to predicting the position of N_i and evaluating the corresponding posterior error before actually deforming the mesh. Using both the reference pose and the previous frame to predict these positions allows, on one hand, to locally return to the reference model when the last frame approximation starts to yield more error than the reference pose, and on the other hand to rely more on the previous frame when tracking error is small. The target position of the control point $\mathbf{t}_i$ which will be used as constraint in the $\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ vectors in Equation 1 is finally defined as the weighted average of all these predictions as

shown in Equation 3.

$$\mathbf{t}_i = \frac{1}{\sum weights}\left[w^0_{i|i}\mathbf{t}^0_{i|i} + w^{t-1}_{i|i}\mathbf{t}^{t-1}_{i|i} + r\sum_{j\in N_i}(w^0_{i|j}\mathbf{t}^0_{i|j} + w^{t-1}_{i|j}\mathbf{t}^{t-1}_{i|j})\right] \quad (3)$$

where r is a coefficient balancing the weight of the neighbours in the computation, and $\frac{1}{\sum weights}$ is normalization using the sum of all applied weights.

3.3. Coarse to fine approach

Experiments confirmed that the range of recoverable deformations is limited when using a fixed radius for the neighbourhood, and thus a roughly constant number of control points. Refining the deformation by progressively reseeding denser sets of control points is much more efficient at recovering finer details. On meshes representing a human, we used three level of details defined by the average number of control points they created on the mesh: 12, 40 and 180. However, 12 random control points are definitely not enough to deform the mesh from its position at time 0 to a decent approximation at time t, and also means loosing most of the information recovered at time $t-1$. Thus at the lowest level of resolution, the mesh is deformed with respect to the result of the previous frame. This allows to roughly align the approximation for time $t-1$ with the observed data at time t, before returning to deforming the mesh from its initial pose with 40 then 180 control points.

4. Results

We performed both qualitative and quantitative analysis of our method on a number of data sequence which were available to us. In the reminder of the section we discuss in details our experimental evaluations.

4.1. Qualitative experiments

We tested our approach using the data provided by J.Starck and A.Hilton who acquired it within the Surfcap project[17]. We present our results on the 5 sequences that were made available to us: Flashkick, Kickup, Head, Lock and Pop. Each of these sequences consists of a Hip Hop dancer performing fast moves that create large deformations and topology changes with respect to the initial surface. We used the reconstructed photo consistent geometry from Starck and Hilton's graph cut method as target point clouds for our method. As these meshes had each more than 100k vertices, we down-sampled them to create set of points and normals with approximately 10k vertices. The deformed reference surface was also a down-sampled version of the first frame of each sequence and had roughly 5k vertices. All the sequences were run using the presented method. The approximated surface for time $t-1$ was first deformed using 12 control points to align it with the point cloud at time t. Then the regular algorithm deforming the reference surface was run using 40 then 180 control points.

We show the results of our method on the five sequences in Figures 4 and in the supplementary video. The top row of each subfigure shows the deformed reference mesh overlaid as wireframe on the original image data from one of the camera of the system, showing that although silhouette constraints are not explicitly enforced, the deformed mesh correctly reprojects in the images. In the bottom rows, rendered versions of the same deformed reference mesh are shown .

4.2. Quantitative experiments

In this section we present a quantitative analysis of the results obtained by our method. Our analysis is based on the measure used by Aguiar et al. in [8] and consists of computing the silhouette reprojection error normalized by the size of the original silhouette. As our data can exhibit severe topology changes, leaving some parts of the deformed template unmatched, this measure is more pertinent than looking for a residual distance between the deformed mesh and the target geometry. Even though we do not explicitly force the surface to fit the silhouettes, we show in Figures 5 that this silhouette re-projection error stays reasonably constant and low over the entire sequences. In the associated video we show the output as the overlay of the mesh structure of the deformed template on top of the original images from one of the cameras.

To demonstrate the contribution multi-resolution approach brings to the final results we depict in Figure 6 the difference between the results obtained when using a small number of control points ($\sim$ 30) and the results obtained when the multi-resolution approach is applied. As the reference pose has a bent left knee, using a coarse control structure only results in the unwanted preservation of this fold, while the coarse-to-fine approach correctly fits the target mesh. The same measure of mean camera silhouette re-projection error confirms significant improvements.

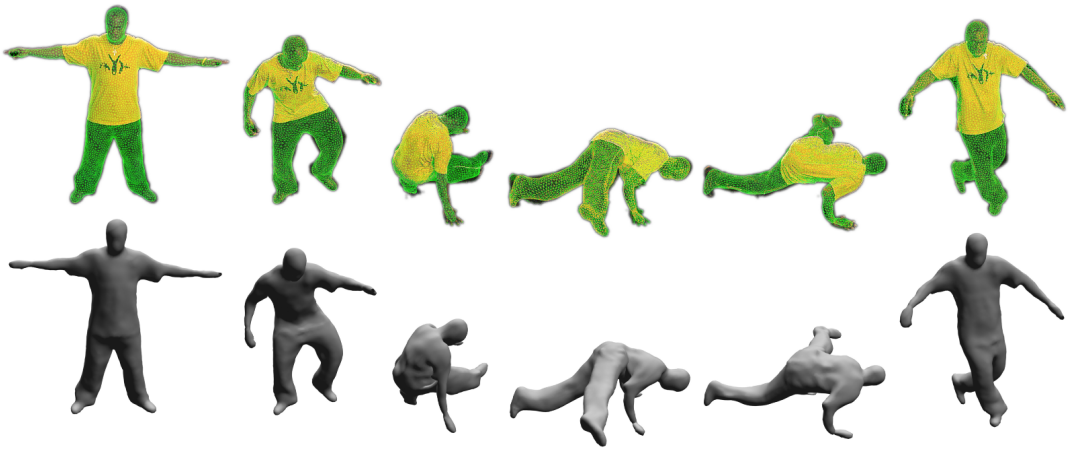
4.3. Discussion

Although our method has performed consistently well on these data sets, with the same set of parameters and no manual intervention, we would like to discuss its limitations and failure cases.

The first limitation is the inability to handle arbitrary *topology changes*. This problem is inherent to the use of a reference mesh as model and is shared by most other approaches, with the notable exception of [19] who handles them at the cost of long term accuracy in the tracking. Self-intersections can to some extend be handled, as they only re-

(a) Head

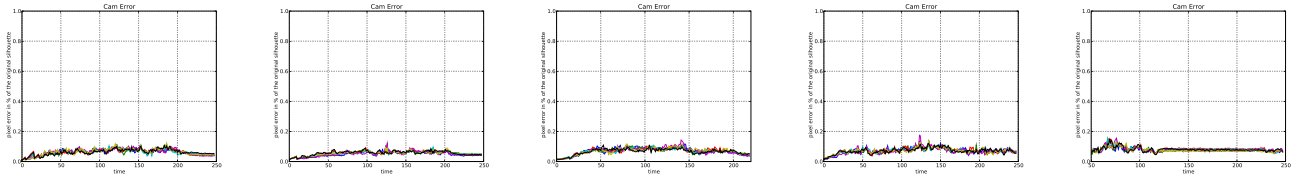
(b) Lock



(c) Kickup

Figure 4. Frames from the Head, Lock and Kickup sequences. The top rows of each subfigure show the overlay of the green wireframe version of the deformed mesh we recover on the original image from the second camera. The bottom rows show the deformed mesh alone.

(a) Silhouette reprojection error: From left to right: original image, silhouette, deformed template, silhouette error.



(b) Lock (c) Pop (d) Kickup (e) Head (f) FlashKick

Figure 5. Silhouette reprojection error as a percentage of the original silhouette: each colour represents a camera. Note that the reprojection error stays small and stable in spite the fact that we do not force explicitly silhouettes to be fitted.

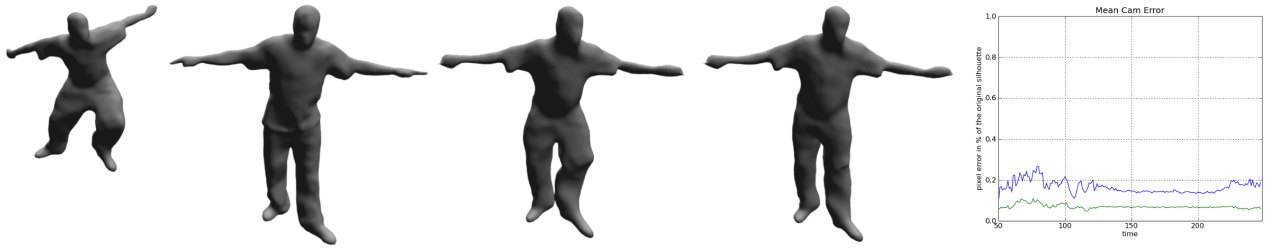


Figure 6. Improvements brought by the coarse to fine approach on the Flashkick sequence. From left to right: initial reference mesh; target mesh; deformed mesh with used $\sim$ 30 control points(note that the left knee is bent following the constraint form the reference mesh model); deformed mesh with the proposed multi-resolution method(the prior imposed by the reference mesh was suppressed by the multi-resolution strategy); silhouette re-projection error for 30 control point case, shown in blue, and the error in case of multi-resolution, shown in green, demonstrate that using control points at multiple resolutions results in more precise capturing of the surface deformations.

sult in some part of the model being temporarily unmatched in the observation. However, the splitting or creation of geometry that happens when a hand gets out of a pocket for example can not be addressed with our approach.

The second limitation lies in the *iterative nature* of our algorithm. Our method basically iteratively fits the input point clouds with a reference mesh. If there are very large errors in one frame of the input geometry, the mesh can be deformed to a state from which the tracking won't be able to recover. However, the neighbour prediction mechanism helps to recover from localised reconstruction errors by propagating information from well matched parts of the deformed mesh to erroneous areas. Accounting for noisy input data constitutes a direction for future work.

Finally, the computational cost of our method is completely dominated by the neighbour prediction and weighting mechanism. For each predicted patch position, a nearest neighbour search is conducted for each of its vertices in the target geometry. As our implementation did not use any space partitioning strategy, this was quite long and resulted in computation times of roughly 5min/frame.

5. Conclusion

We presented a method for capturing the dynamic evolution of a surface from independent multi-view surface reconstructions. The object reconstructed in the first frame is used as reference model and fitted to the points and normals of the independent reconstructions obtained at each time frame. Dealing with fast motion, large deformations and topology changes makes it difficult to extend robustly the ideas of ICP. We propose a novel algorithm that meets this challenge and performs consistently well in various data sequences. It is based on randomly seeded control points around which rigid motions are locally averaged. Three main contributions are to be distinguished in the overall approach: the accounting for point dynamics when establishing model to data correspondences, the higher level rigidity model between the control points and the coarse-to-fine strategy that allows to fit the temporally inconsistent data more precisely.

Acknowledgements

This research was funded by *Deutsche Telekom Laboratories* and partly conducted in their Berlin laboratory.

References

[1] N. Ahmed, C. Theobalt, C. Rössl, S. Thrun, and H.-P. Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. In *CVPR*, Anchorage, Alaska, 2008. IEEE Computer Society. 2

[2] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM SIGGRAPH*, pages 587–594, New York, NY, USA, 2003. ACM. 2

[3] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007. 2

[4] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. 2

[5] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.*, 26(3):33, 2007. 2

[6] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. In *ACM SIGGRAPH*, pages 1–9, New York, NY, USA, 2008. ACM. 2

[7] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 1991. 2

[8] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):1–10, 2008. 1, 2, 5

[9] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *IJCV*, 18(18):99–119, 1996. 2

[10] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 1–8, Miami, USA, 2009. IEEE Computer Society. 1, 2

[11] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *SIGGRAPH: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, New York, NY, USA, 2001. ACM Press. 2

[12] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987. 4

[13] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003. 4

[14] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, 2006. 1

[15] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007. 2, 4

[16] J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *ICCV07*, pages 1–8, 2007. 2

[17] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007. 2, 5

[18] C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, and H.-P. Seidel. Template deformation for point cloud fitting. In M. Botsch and B. Chen, editors, *Symposium on Point-Based Graphics*, pages 27–35, 2006. 2

[19] K. Varanasi, A. Zaharescu, E. Boyer, and R. P. Horaud. Temporal surface tracking using mesh evolution. In *Proceedings of the Tenth European Conference on Computer Vision*, volume Part II of *LNCS*, pages 30–43, Marseille, France, October 2008. Springer-Verlag. 2, 5

[20] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH*, pages 1–9, New York, NY, USA, 2008. ACM. 1, 2