HAL
open science

Iterative Mesh Deformation for Dense Surface Tracking

Cédric Cagniart, Edmond Boyer, Slobodan Ilic

To cite this version:

Cédric Cagniart, Edmond Boyer, Slobodan Ilic. Iterative Mesh Deformation for Dense Surface Tracking. 3DIM 2009 - IEEE 12th International Conference on Computer Vision Workshops, Oct 2009, Kyoto, Japan. pp.1465-1472, 10.1109/ICCVW.2009.5457440 . inria-00440389

HAL Id: inria-00440389

https://inria.hal.science/inria-00440389

Submitted on 23 Feb 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Iterative Mesh Deformation for Dense Surface Tracking

Cedric Cagniart
CAMP TU Munich
& Deutsche Telekom Laboratories
cedric.cagniart@in.tum.de

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

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

Abstract

In this paper we present a new method to capture the temporal evolution of a surface from multiple videos. By contrast to most current methods, we introduce an algorithm that uses no prior of the nature of tracked surface. In addition, it does not require sparse features to constrain the deformation but only relies on strictly geometric information : a target set of 3D points and normals. Our approach is inspired by the Iterative Closest Point algorithm but handles large deformations of non-rigid surfaces. To this end, a mesh is iteratively deformed while enforcing local rigidity with respect to the reference model. This rigidity is preserved by diffusing it on local patches randomly seeded on the surface. The iterative nature of the algorithm combined with the softly enforced local rigidity allows to progressively evolve the mesh to fit the target data. The proposed method is validated and evaluated on several standard and challenging surface data sets acquired using real videos.

1. Introduction

Capturing the motion and deformation of object surfaces is a fundamental task when analysing dynamic scenes using several cameras. In essence, this task relies on the ability to build shape models from multiple views as well as to identify the motion of such models over time sequences. While numerous computer vision methods solve for the first issue and allow precise photometric models of objects to be build using multiple static views[12], less consider the second issue and provide motion information. However motion cues are required by many applications that analyse or duplicate shape evolution over time as, for instance, body motion capture applications.

One strategy to tackle surface tracking is to establish a succession of mappings between shapes independently reconstructed at adjacent time frames. This can be done by diffusing the information brought by sparse visual and geometric features[14, 1, 16]. The interest is to be able to estimate motion information without any assumption on the observed shape. On the other hand, this strategy fails to provide a consistent model over time unless a dense correspondence function is identified, which appears to be difficult to achieve in a robust manner over several frames.

Figure 1. Results on the Kickup sequence[15]. We show a coloured version of the deformed mesh. The fact that the colouring stays consistent over the sequence indicates dense tracking.

Another strategy consists in evolving a reference surface with respect to the shape information obtained over time sequences. In that case, the surface representation is, by construction, temporally consistent. Recent works, e.g. [8, 17], have demonstrated the efficiency of this approach to robustly model complex surface evolutions, although the reference surface limits the range of possible deformations.

We adopt a similar strategy in this paper and propose to track a deforming mesh using independent reconstructions obtained with a multiple camera set-up. In contrast to the afore mentioned works, our method does not assume a parametric motion model nor requires specific features, e.g. photometric features, both being not necessarily available in many practical situations. Instead, we directly estimate the motion and deformation from purely geometric information as obtained by the acquisition system. Our approach builds on the Iterative Closest Point algorithms[4] and iteratively re-estimates point to point associations using surface point locations and surface normals. The current surface is then deformed with rigid motions estimated locally by averaging point displacements on randomly seeded patches. The local rigidity of the deformation is ensured by two mechanisms which operate at different levels of details.

At the finest level, we use the preservation of local differential coordinates which has been successfully used in

2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops
978-1-4244-4441-0/09/$25.00 ©2009 IEEE

interactive modeling applications[13]. In our case, the deformation must be constrained automatically. We therefore introduce for robustness an other rigidity constraint at the coarser patch level that diffuses locally the computed rigid motions between neighbouring patches. As a result, the algorithm finds a compromise between surface observations and local rigidity constraints and the deformed surface only approximates the surface observations, hence being more robust to corrupted data sets. These data sets can come from various acquisition systems including multi-view stereo or shape from silhouette. While simple the proposed scheme appears to be efficient and versatile as demonstrated in this paper with various scenarios.

The remainder of this paper is organised as follows : in section 2 we discuss other approaches to our problem. In Sections 3 and 4 we present our algorithm and our results.

2. Related Work

As stated in the introduction, our algorithm tackles surface tracking by evolving a reference mesh over time. To this aim, it iteratively deforms the reference mesh and re-estimates point-to-point correspondences between the mesh vertices and a set of target points. In that respect it is inspired by the Iterative Closest Point algorithm[7, 4] that was initially proposed to register rigid motions of solid objects. While extensions of this work to non-rigid deformations have been proposed, e.g. [9, 2, 3], to the best of our knowledge none applies to the case of large deformations as observed when capturing body motions. Over the existing methods that address this issue, two main classes can be identified in connection with our approach:

Dense matching from sparse features The traditional way to match two surface points is to consider two-dimensional image interest point descriptors such as SIFT, among others. Unfortunately such interest points are usually non-uniformly distributed on the surface, hence giving little information on how textureless regions deformed. Different solutions have been proposed to obtain more homogeneously distributed sparse matches. For instance Naveed[1] propagates the sparse features information on the rest of the mesh using level sets of harmonic functions.

Other visual features, such as edges, are used by Starck and Hilton[15, 14]. In [6], a similar problem of markerless garment capture is addressed by using the boundaries of the garment as anchors to guide the establishment of a consistent cross-parametrization between the independently reconstructed surfaces.

Geometric features have also been used. Varanasi *et al.* [16] match mesh extremities identified as the extrema of the geodesic integral [10]. Starck and Hilton [14] add uniformly distributed geometric features, in the form of the geodesic-intensity histogram, and then regularize the assignment using a MRF on the graph of the mesh. These geodesic features require special care when topology changes appear, as they loose the reliability one expect from sparse features.

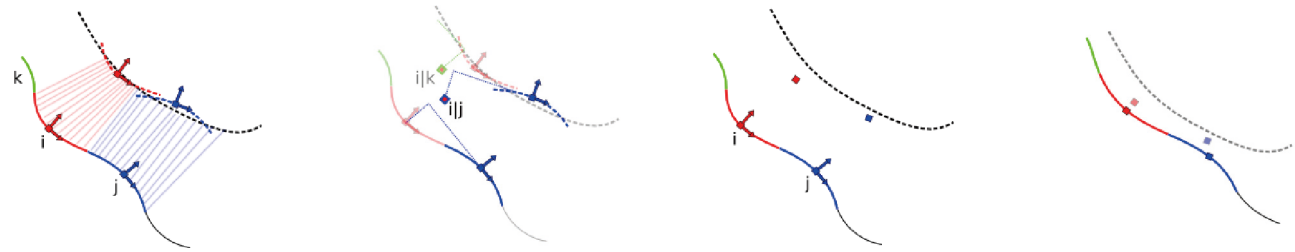
Our work avoids the need for such features and focuses on how to fit a mesh with a set of target points.

Tracking by deformation This class of approaches requires rigidity priors that guide the surface deformation and improve its robustness. The particular case of human shape and motion has received a lot of attention. Several methods in this category perform the deformations of an initial high resolution mesh template, obtained using a laser-scanner for instance [5, 8, 17]. In [8, 17], and closely related to our work, silhouette constraints are diffused over the mesh by enforcing the preservation of Laplacian coordinates. In [8], a coarse volumetric mesh is first deformed, then higher frequency deformations are estimated locally. In [17] a pose is first estimated by fitting a skeleton model and then the corresponding shape estimation is then refined by inflating the surface in order to match the silhouettes. Interestingly, the method presented in [16], based on the preservation of differential coordinates, is able to handle topology changes by performing a mesh morphing step after the deformation of the surface. Unfortunately, the lack of rigidity constraint in the morphing stage prevents long term accuracy.

We propose a simple but efficient way to enforce the local rigidity of a surface. First, fine details are kept by preserving differential coordinates. Second, larger deformations are limited by a lower resolution spring like force applied to sparse points seeded independently at each time step. Our contribution with respect to existing approaches is to provide rigidity constraints that are surface based and that are localized both in space and time. As shown in the paper, the associated deformation scheme allows to track surfaces in standard data-sets without the need for precise prior models nor for specific features. In addition, we believe that this scheme allows for further explorations of the problem that include topology changes.

3. Method

Our approach belongs to the *tracking by deformation* class of methods that deform a reference surface over time. In a way similar to the ICP algorithm, the method iteratively re-estimates point-to-point correspondences between the target point cloud and the current approximation of the deformed mesh. This assignment procedure is detailed in 3.1. Using the reference surface as a model reduces the impact on the tracking of corrupted data and wrong assignments that appear when dealing with highly deformable surfaces. In the framework we use, the surface deformation is achieved by setting position constraints on some vertices of the mesh. The evolved mesh is obtained by finding a com-



(a) Averaging rigid motions on patches.

(b) Computing inter-patch constraints : the target patch positions knowing the neighbouring rigid motions.

(c) Computing final targets for the patch centers.

(d) Use these targets as soft constraints to the deformation framework.

Figure 2. Method Outline

promise between these soft-constraints and the preservation of the position of each vertex with respect to his neighbouring vertices. This could seem to be a sufficient way of enforcing local rigidity. However, as show in 3.2, the way this framework handles error on the constraints is not sufficient. This motivates the need for a a higher level rigidity model, as introduced in 3.3 which complements the first by adding a rigidity force between the constraints themselves. To help the reader go trough the next paragraphs we give an outline of the method Algorithm 1. The inner loop of the algorithm is illustrated in Figure 2.

Algorithm 1 Method Outline

$\hat{X}^0 = X^0$
for t in timeFrames **do**
  Seed patches randomly on the surface
  while the patch centres are moving **do**
    Compute a target position field.
    Average the rigid motions on patches.
    Compute inter-patch constraints.
    Compute the final targets for the patch centres.
    Use them as soft constraints in the mesh deformation framework.
  end while
  Output $(M, \hat{X}^t)$ as approx for time t
end for

3.1. Computing target positions for the vertices

As our method deforms a reference mesh across the sequence without changing its topology, we are actually keeping the same connectivity (ν, τ) where ν is the set of vertices and τ is the set of triangles. We are only re-evaluating a position function $\hat{X} : \nu \mapsto \mathbb{R}^3$ mapping vertices to their cartesian coordinates.

The classical ICP algorithm looks for the rigid motion minimizing a distance function between a solid model and the observed data. Another point of view is to consider that the *closest compatible point search* suggests a position function on ν and that the *rigid motion estimation* that follows is providing a regularization of the function by forcing the deformation to be a direct isometry.

Our closest compatible point search is very simple as we declare a point of the target set and a point of the deformed mesh compatible if their normals form an angle smaller than $45°$. However we proceed in a slightly different way : instead of having the vertices of the deformed mesh look for the closest point in the target set, we go through the target set and have each of the vertices contribute to the position of its closest compatible point in the deformed mesh. This means that we are not forcing the function to be defined densely on ν. The weight of the contribution is simply the dot product of normals. This preserves the contribution of unmatched parts of the surface even if they are still far away and is therefore more efficient at getting out of local minima, although it also increases our sensitivity to errors in the target point set.

Our approach is motivated by the fact that the regularisation of the deformation field and therefore of the position function can be done locally on the mesh. In our case the notion of locality involves using the connectivity of the graph τ which differentiates us from the regular ICP which only consider point clouds.

3.2. Mesh Deformation Framework

Among the mesh deformation methods which have been developped in the computer graphics community, the preservation of local differential coordinates has proven to be a reliable and easy way to deform surfaces using only a reference mesh. It was also identified as a powerful tool for computer vision, particularly for the specific problem we address[8, 16, 17].

The reference mesh defines an initial position function X^0 which can be written as three $|\nu| \times 1$ vectors x^0, y^0, z^0 containing the cartesian coordinates. The idea is to build the laplacian matrix L from (ν, τ), weighted with cotangent

weights[11] computed with X^0. It is then possible to define a rigidity energy for each of the cartesian coordinates x, y, z independently (here for the x coordinate) :

$$E_r = \|Lx - Lx^0\|^2 \tag{1}$$

Solving for x using this constraint only yields an under-constrained linear system of equations. Given a set of vertices whose positions we wish to constrain, we can define a diagonal weight matrix W_c (with weight 0 if the vertex is un-constrained) and three vectors x_c, y_c, z_c containing the target cartesian coordinates for the constrained vertices. We are then solving for each coordinate independently a system :

$$E_r + E_c = \|Lx - Lx^0\|^2 + \|W_c(x - x_c)\|^2 \tag{2}$$

Provided we have at least one constrained vertex, this yields a full-rank system in the least-squares framework. But this approach has problems as soon as we work with large deformations. If, for example, we rotate the surface by $180°$ with respect to the reference pose, E_r will push the vertex in the wrong direction. In recent works by Sorkine[13], an iterative scheme is presented to evaluate local rotations of the surface, which allows to bring a degree of rotation invariance to the process. Taking into account the re-estimated local rotations and the constraints, the idea is to then minimize the following energy the least-square sense:

$$\begin{aligned} E_r + E_c = \|Lx - R(Lx^0, Ly^0, Lz^0)\|^2 \\ +\|W_c(x - x_c)\|^2 \end{aligned} \tag{3}$$

The method presented in the paper solves iteratively for x and the function R. It is as advertised easy to implement and efficient in that the costly Cholesky factorisation of the big sparse matrix $L^T L + W_c^2$ only happens once for a given W_c matrix.

The idea of setting dense constraints on the mesh and letting the least-square procedure de-noise the dense position field is tempting. From equation 3 we understand that a choice of $W_c = I$ would give the preservation of local differential coordinates and the constraint coordinates the same importance. If we assume that the observation noise on our set of vertices is centred, uncorrelated and gaussian, the least square minimization performs extremely well as shown in Figure 3. Unfortunately, computing a target position field by looking for the closest compatible point yields un-centred and highly correlated vertex position errors.

3.3. Sparse constraints and high level rigidity

These last remarks on the position error brought by the closest compatible point search have motivated our approach. Instead of constraining each vertex to its target position, we seed patches on the mesh with a maximal patch radius. We compute for each patch centre a target position by averaging and diffusing the rigid motions on the patches. The resulting sparse set of *soft constraints* on the patches centres is then used to control the mesh deformation.

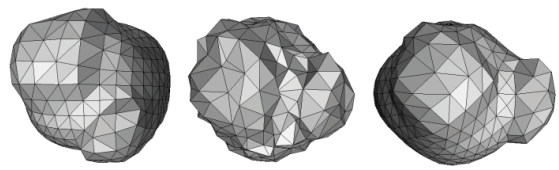


Figure 3. The original mesh (left) is rotated and translated then added to zero meaned uncorrelated noise (centre). Minimizing the least square error on differential coordinates allows to recover the rigid motion correctly (right)

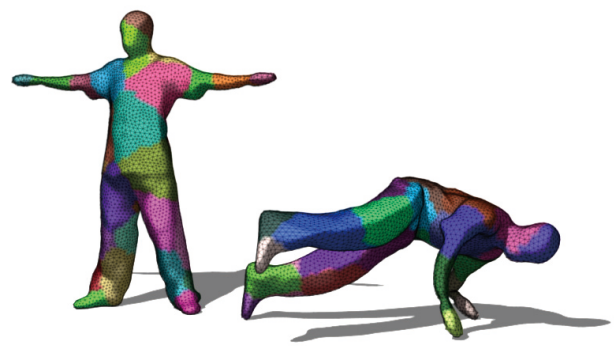


Figure 4. Patches are randomly seeded on the surface to deform. The first mesh is the initial pose. The second one is a deformed version of the first, patched differently because the patching process happens independently for each frame of the sequence.

The patch seeding takes place at each time step independently. This means that the patching procedure is not part of our model. This helps in keeping localised the computation of constraints not only spatially but also temporally. The process starts by distributing patch centres on the extremities, which are local maxima of the geodesic integral defined from the geodesic distance as in [10]. The rest of the patches is distributed randomly until the whole graph is covered. An example of patchings for two different frames of a same sequence is presented in Figure 4.

First we estimate a rigid motion of the patches toward their associated target data points. It is done by computing an average of direct rigid transformations of the vertices to their assigned target positions. To do so we use the same technique[13] as the one we use to compute local rotations of the surface with respect to the reference pose in the laplacian deformation framework. The residual error of this rigid motion is a first measurement of the quality of the future alignment and is therefore used to weight the associated constraint in the deformation step.

Because patch centres often lie pretty close to the centre

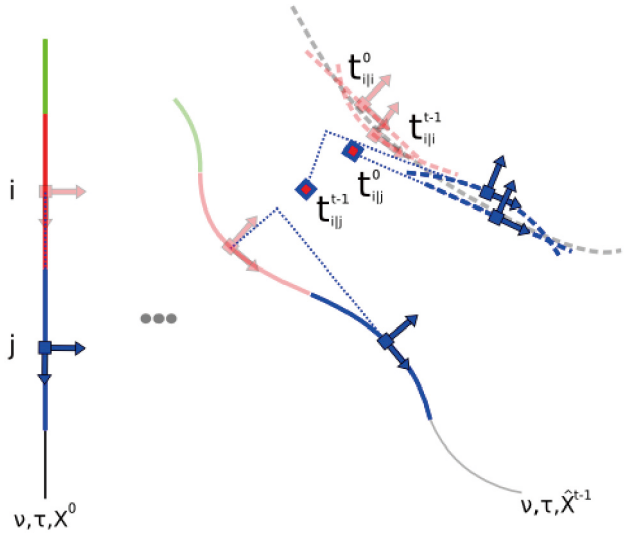


Figure 5. The rigid motion is diffused between neighbouring patches.

of gravity, the closest point force does not provide much rotation information to the deformation step alone. Enforcing a local rigidity model ensures robustness to point assignment errors and diffuses rigid motions and their certainties at a coarser level of detail : the patch level. Instead of simply computing the target position for the patch centre c_i using the rigid motion of the corresponding patch i, we also use the rigid motions from the neighbouring patches $j \in N(i)$ to predict the target position. Basically, for the patch i we look for the information : "Where would my neighbours think I should be if we were a rigid transformation of (ν, τ, X^{t-1}) or $(\nu, \tau, \hat{X}^0)$?".

On Figure 5 we show that after computing the rigid motion for patch j between $t-1$ and the target positions, we can predict the position of the centre of patch i by assuming that the rigid motion was the same for him. This way we obtain a target $t^{t-1}_{i|j}$. The exact same can be done by computing the rigid motion for patch j from the reference mesh to the target positions, yielding another target $t^0_{i|j}$. With the same notation we define $t^0_{i|i}$ and $t^{t-1}_{i|i}$ as the targets obtained by using the rigid motions computed for the patch i itself. We can finally set the final target t_i as a weighted average of all these targets.

$$t_i = \frac{1}{w_i + r\sum w_j}[\alpha(w_i.t^0_{i|i} + r\sum_{Pj\in Ni} w_j.t^0_{i|j}) \\ +(1-\alpha)(w_i.t^{t-1}_{i|i} + r\sum_{Pj\in Ni} w_j.t^{t-1}_{i|j})]$$

The r term controls the rigidity as it weighs the importance of the neighbouring rigid motions in the computation of the target. The α term balances the influence of the previous and reference frames. In practice $\alpha = 0.4$ was used for all the results the results which we present in the next section.

4. Results

We show in this section the performance of our algorithm in different video sequences involving large surface deformations. In the following paragraphs we present our results on six sequences. Four of them were computed from reconstructed photo-consistent meshes. The two others involved noisy visual hulls computed with a very simple voxel carving algorithm.

4.1. Using Independently Reconstructed Meshes

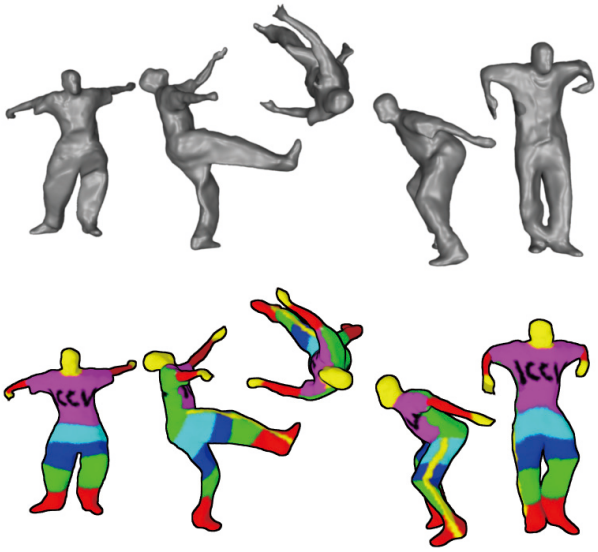
Our method was initially tested using independently reconstructed meshes from which we extracted 3D point and normal data. The idea was to deform a low resolution and smooth template using high resolution and uniformly sampled target point sets.

Challenging sequences were provided by the Surfcap project of J. Starck and A. Hilton [1] as a set of very smooth reconstructions obtained with a graph-cut method[15]. As the given geometry was at a extremely high resolution (more than 100k vertices per mesh) we down-sampled the target meshes to 10k vertices. The deformed reference model was a down-sampled version of the first frame with roughly 5k vertices. All the sequences were run with the same set of parameters. The Figure 4 show the size of the patches on two frames from the Kickup sequence. There were were usually 23 to 28 patches seeded on the mesh and thus as many control points used to guide the deformation. The Kickflip sequence shown in Figure 6(a) presents a hip-hop dancer performing a move exhibiting both large deformations and very fast movement. Out of the 250 frames we ran 150 because the 50 firsts had one of the hands attached to the leg and therefore were not suited as models. Head, Pop and Lock were the other sequences we ran. For them we used frame 0 as reference and show in the corresponding Figures 6(b), 6(c) and 9 that our approach did recover a very wide range of large deformations. For each figure, we show on the top line the the target meshes which are not temporally consistent. On the bottom line we show a coloured version of the deformed mesh. The consistence in the colouring indicates dense tracking.

4.2. Using Visual Hulls

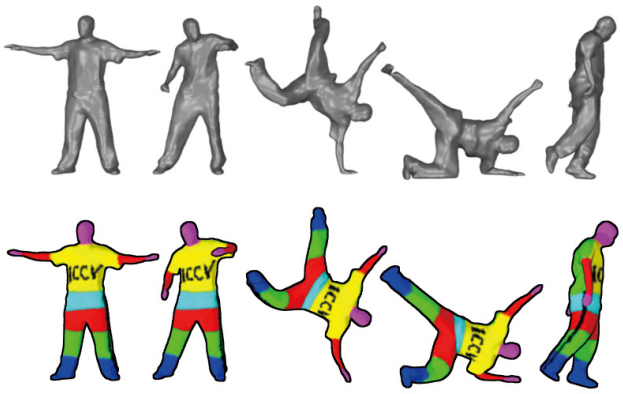
We tested our approach against two of the datasets made publicly available by Daniel Vlasic[2] from MIT. Both

[1] http://www.ee.surrey.ac.uk/CVSSP/
[2] http://people.csail.mit.edu/drdaniel/



(a) Kickflip results

(b) Lock results



(c) Head results

Figure 6. Results on the Starck and Hilton sequences. Shaded models on the top lines are the independent reconstructions. On the bottom lines we show coloured versions of the deformed meshes computed by our algorithm. The consistence in the colouring indicates dense tracking.

datasets are 174 frames long. We fed the calibration and silhouette data into a very simple voxel carving algorithm to generate meshes whith roughly 10k-15k vertices.

The laser scanned reference meshes provided with the datasets were extremely detailed but had too many vertices and high frequency folds. We created sub-sampled and smoothed out versions of these templates with approximately 3.5k vertices. We show in Figure 7 the results on the sequences Handstand and Crane. On the top line we show the independently reconstructed visual hulls. On the bottom line we show the reference surface deformed along the sequence. Here again the consistency of the colouring shows the density of our tracking.

Although the 2D segmentation was pretty close to perfect, the meshes obtained by voxel carving were still noisy. The noise from shape from silhouette techniques is hard to handle. It does not create small variations around the real surface but creates fake volumes or even holes. In terms of points and normals, this means we have a lot of completely erroneous points pulling on the deformed mesh. Our algorithm still managed to process the sequences but the quality of the obtained deformations is directly related to the quality of the input data.

4.3. Quantitative evaluation

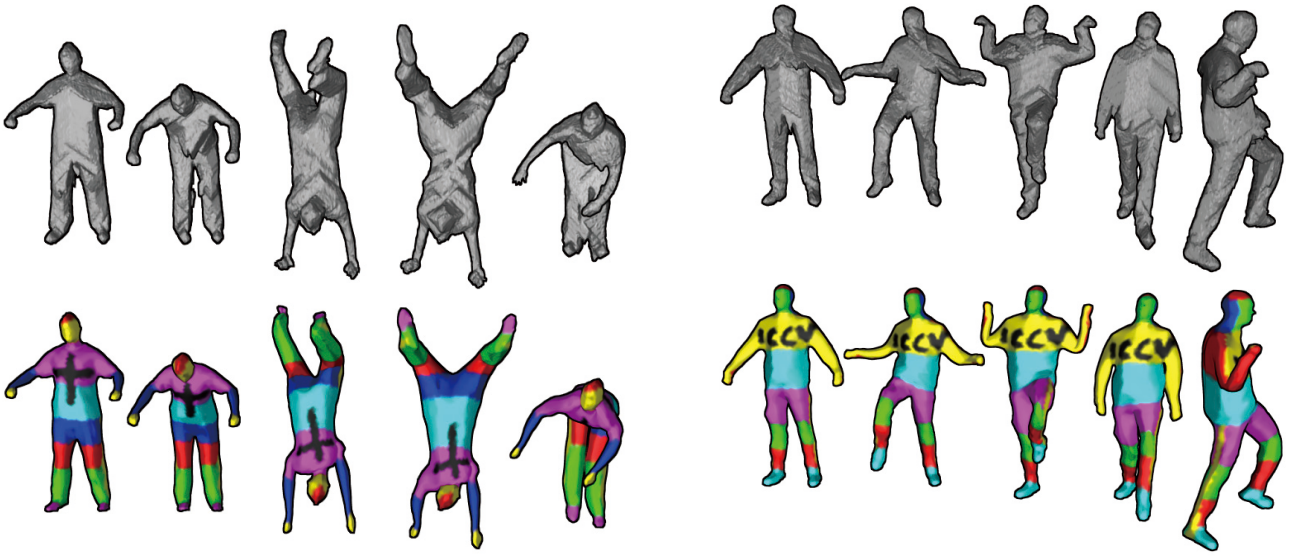
To evaluate our results qualitatively, we used the silhouette reprojection error which is the proportion of pixels that either project outside of the original silhouette when they should be in, or project inside when they should be out. We present in Figure 8(a) the evolution of this error in the pop sequence. It must be noted that these results are obtained without explicit fitting of the 2D silhouettes.

It is possible to apply a post processing step at each frame where the deformed result mesh is further deformed smoothly with the same framework, but this time such that the silhouette reprojection error is minimized. This is a very local procedure and works only because the deformed mesh is already very close to the solution. We present numerical results in Figure 8(b) to be compared with Figure 8(a).

4.4. Performance and Shortcomings

On the Kickup sequence, the average frame processing time which was 4.8 seconds on a 2.5Ghz quad-core processor. The frame rate was constant throughout the sequence which exhibits large deformations but no major ambiguity.

Unsurprisingly, our approach encounters problems when confronted with ambiguous situations such as these brought by topology changes such as self-intersections. The tracking can get locally lost for some time but the rigidity constraints with respect to the reference model allow it to recover once the ambiguity disappears. We give an example of such a situation in Figure 9. Our algorithm gets lost shortly after frame 120. This corresponds to a subsequence



(a) Handstand results (b) Crane results

Figure 7. Results on the Vlasic sequences. The visual hulls are shown on the top lines. On the bottom lines we show coloured versions of the deformed meshes computed by our algorithm. The consistence in the colouring indicates dense tracking. Note that the quality of the recovered deformation is reasonable considering the quality of the input data.

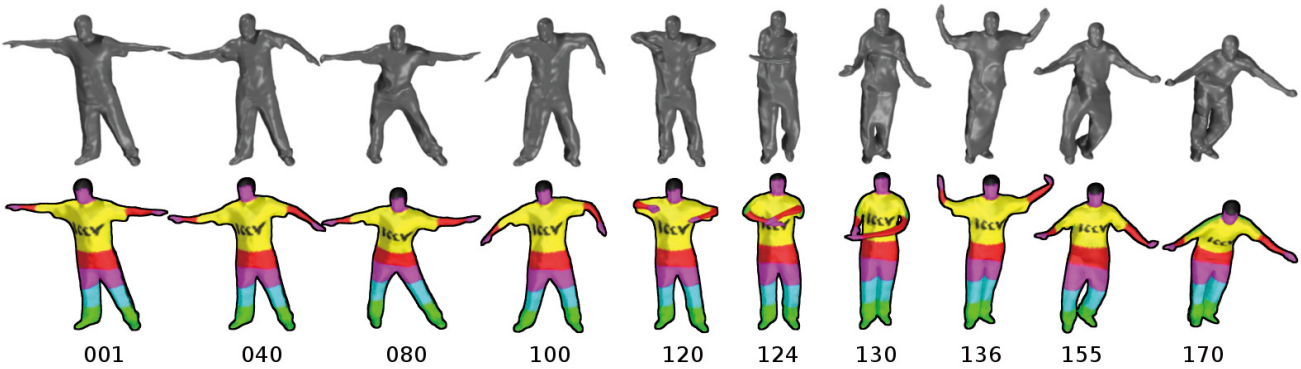
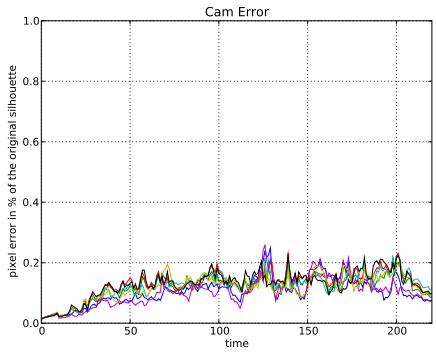


Figure 9. Heavy topology changes create ambiguity around frame 130 but the tracking is recovered correctly at frame 136.

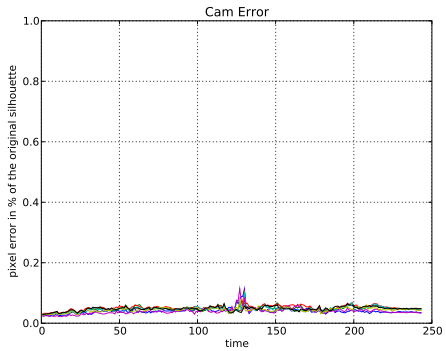
where the dancer crosses his arms and where the independent reconstructions create fake volumes between the arms and the body. However, a couple of frames later our algorithm recovers because it uses the residual error on the patches rigid motions approximations to weight the constraints. This means that a high residual error will relax the corresponding constraints and let the mesh go back locally to the reference position from which it finds better point-to-point associations.

5. Conclusion

We presented a method for dense surface tracking by iteratively deforming a reference mesh along a temporal sequence. Our approach relies on nothing but on a set of points and normals obtained from static 3D reconstruction techniques. We iteratively deform the surface to progressively fit the target point cloud. We ensure local rigidity at a fine level by using a mesh deformation framework which preserves local differential coordinates. We introduced a simple way of enforcing rigidity at a coarser level, by seeding patches on the surface and diffusing the rigid motion estimations between neighbours. Our experiments on many



(a) These results are obtained without explicit fitting of the 2D silhouettes.



(b) After post-processing deformation to enforce silhouette consistency the error is considerably lower.

Figure 8. Silhouette reprojection error in the 7 cameras on the pop sequence.

datasets show the efficiency our approach and encourages further exploration.

Acknowledgements

This research was funded by Deutsche Telekom Laboratories. We would like to thank Jonathan Starck, Adrian Hilton (U. of Surrey) and Daniel Vlasic (MIT) for giving us access to the datasets.

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] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 587–594, New York, NY, USA, 2003. ACM.

[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.

[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.

[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.

[6] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM.

[7] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. pages 2724–2729 vol.3, 1991.

[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.

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

[10] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, New York, NY, USA, 2001. ACM Press.

[11] 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.

[12] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. volume 1, pages 519–528, 2006.

[13] 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.

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

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

[16] 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.

[17] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM.