Free-Form Mesh Tracking: a Patch-Based Approach

Cedric Cagniart, Edmond Boyer, Slobodan Ilic

Free-Form Mesh Tracking: a Patch-Based Approach

Cedric Cagniart[1], Edmond Boyer[2], Slobodan Ilic[1]

[1]Department of Computer Science, Technical University of Munich
[2]Grenoble Universités, INRIA Rhône-Alpes
{cagniart,slobodan.ilic}@in.tum.de, edmond.boyer@inrialpes.fr

Abstract

In this paper, we consider the problem of tracking non-rigid surfaces and propose a generic data-driven mesh deformation framework. In contrast to methods using strong prior models, this framework assumes little on the observed surface and hence easily generalizes to most free-form surfaces while effectively handling large deformations. To this aim, the reference surface is divided into elementary surface cells or patches. This strategy ensures robustness by providing natural integration domains over the surface for noisy data, while enabling to express simple patch-level rigidity constraints. In addition, we associate to this scheme a robust numerical optimization that solves for physically plausible surface deformations given arbitrary constraints. In order to demonstrate the versatility of the proposed framework, we conducted experiments on open and closed surfaces, with possibly non-connected components, that undergo large deformations and fast motions. We also performed quantitative and qualitative evaluations in multi-cameras and monocular environments, and with different types of data including 2D correspondences and 3D point clouds.

1. Introduction

Recovering the temporal evolution of a deformable surface is a fundamental task in computer vision, with a large variety of applications ranging from the motion capture of articulated shapes, such as human bodies, to the intricate deformation of complex surfaces such as clothes. Methods that solve for this problem usually infer surface evolutions from motion or geometric cues. This information can be provided by motion capture systems or one of the numerous available static 3D acquisition modalities. In this inference, methods are faced with the challenging estimation of the time-consistent deformation of a surface from cues that can be sparse and noisy. Such an estimation is an ill posed problem and prior knowledge must be introduced in order to limit the range of possible deformations. Existing methods differ in the priors they consider.

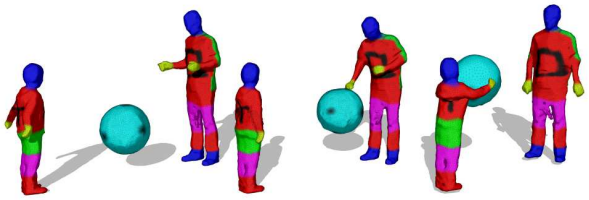


Figure 1. Tracking of a complex surface, composed of disconnected components, using multi-view reconstructions.

Many approaches are based on strong shape priors, e.g. parametric models [18, 7] machine learning [4] or modal approaches [14], that reduce the dimensionality of the deformation search space. While increasing robustness, these approaches lack generality and they quickly loose precision when the observations can not be explained by the assumed model, which often occurs in practice. Other approaches make less assumptions and recover the temporal evolution of a surface by computing the deformation of a reference mesh. The problem becomes then an optimization over vertex positions, where prior knowledge acts as a regularization term over vertex displacements. Among the existing schemes, several methods try to enforce physically plausible deformations by simulating resistance to stretching and bending [3, 8]. To this purpose, they usually operate at the vertex level where the noise has a strong influence, hence leading to difficult non-linear optimizations.

Our motivation in this work is to provide a generic data-driven mesh deformation framework that assumes little knowledge on the tracked surface and can hence cover a large range of applications and shapes (see for instance Figure 1). We adopt a strategy that divides the original surface into regions, or patches, to which vertices are attached. These patches are used to locally average data terms and to enforce inter-patch rigidity constraints with respect to a

reference pose. They therefore define an abstraction layer above surface vertices that allows for coarse to fine sampling of the recovered deformation through increasing patch resolutions. This appears to be an important feature that enables to handle, in a unified way, noisy data and large deformations while still recovering local details.

Our work builds on tools developed in the computer graphics community [2, 17, 12] to consistently deform meshes in physical simulations or geometric modelling applications. We extend their use to applications involving noisy data, through our surface patch framework and a robust numerical optimizer. We have successfully applied this method to various scenarios and we demonstrate its ability to adapt to two different contexts: monocular cloth tracking and multi-camera setups.

The rest of this paper is as follows: Section 2 recalls existing works that deal with surface tracking. In Section 3 the proposed framework is presented in more details. Section 4 shows how it performs in different applications. In Section 5, we discuss the issues and limitations of our framework.

2. Related Work

As mentioned before, methods that solve for surface tracking from visual data differ by the prior knowledge they consider about the observed surfaces. This prior knowledge can range from strong priors on shape models to smooth rigidity constraints between vertices on a mesh. Closely related to our work, a large class of approaches deforms a mesh template with various constraints to ensure plausible deformations and fixed topology.

Dimension Reduction Some methods use very strong constraints on the deformations to increase their robustness. While efficient this limits the application domain. In the widely studied case of human motion capture, Vlasic [18] and Gall [7], for instance, optimize the pose of skeletal models before fitting the surface to visual data. Other approaches proceed by first learning deformation modes of the object of interest. Chai [4], in the case of data-driven face animation, tracks a low number of facial features and maps them back onto a linear combination of previously acquired high resolution motion data using machine learning techniques and a database of laser scanned faces. Salzmann [14] recovers the 3D deformation of a piece of cloth by using Principal Component Analysis on a randomly generated set of possible configurations to constrain the space of acceptable deformations and to reduce the optimization complexity.

Mesh Based Regularizations More generic methods use physics-inspired models to ensure as rigid as possible deformations. This is usually achieved by penalizing the change of differential properties of the surface such as edge length or curvature with respect to a rest state [8]. However, as discussed in a recent survey [3] the importance of local rotations of the surface when computing these differential properties makes the surface deformation problem inherently non-linear. This means that although operating at a fine scale allows to recover detailed changes of curvature, increases in mesh resolution can make the associated non-linear optimization computationally impractical. Recently, a number of computer vision works, e.g. [11, 6] used rigidity priors based on the preservation of Laplacian coordinates and on the implicit optimization of the local transformations [15]. Although these Laplacian editing methods behave well for their original purpose, that is as interpolation algorithms between manually set constraints, their use as regularization term for large motions in noisy environments must be handled with care. In fact [18, 7] only use it to preserve the small scale details of the surface once the general motion has been found, while [5] starts by using a variation of the technique on a coarse tetrahedral mesh before using it to solve for small deformations.

Other Regularizations Other works in geometric modelling decouple the complexity of the original geometry from the representation of the deformation by using a coarse control structure. For example, Summers and Li [17, 10] optimize their cost function on a deformation graph, while Botsch embeds the surface in extruded volumetric prisms [1] or in a cubic lattice [2]. However, they avoid linearisation artifacts by embracing non-linearity and optimizing explicitly on the local rotations of the surface.

Our method extends this class of approaches for deformations driven by visual data. Decoupling the original geometry from the representation of the deformation is of particular interest as it allows to use the original vertices as sampling domain for data terms while benefiting from averaging effects when integrating the information on the control structure. Moreover, optimizing explicitly on local transformations of the surface allows a finer integration of the elementary motion cues that are sampled on the vertices.

3. Patch based deformation framework

The presented framework deals with mesh deformation. The reference shape is discretized as a set of vertices and triangles (ν, τ) and defined as a reference position function $\mathbf{x}^0 : \nu \mapsto \mathbb{R}^3$. As this mesh is deformed to minimize an energy function, a position function $\mathbf{x}$ is iteratively re-estimated. However, optimizing directly on these positions can become computationally infeasible as the mesh resolution increases. Moreover, and as discussed in Section 2, the success of methods based on dimension reduc-

Figure 2. Patches with the Stanford armadillo (173k vertices) with a maximum patch radius of 30.

tion demonstrates that the deformation belongs to low dimensional spaces in many applications. Consequently, the mesh should advantageously be seen as a *sampling domain* over which the actual optimization is performed at some higher level: an abstract representation of the shape decoupled from the complexity of the original geometry. Embedding the shape in a lower dimensional control mesh or lattice is a common way to achieve this decoupling.

In this section we describe an algorithm to divide the original mesh into surface cells called *patches*. We present then the associated deformation framework as well as how rigidity priors are encoded on this simplified control structure. In this framework, the optimal local rigid transformations of patches are searched for with respect to various data terms that are sampled and averaged on patches. These data terms are application dependent and details are given in the section 4 along with experimental results.

3.1. Patches

Surface patches define a shape representation that should ideally follow the intrinsic structure of the shape, e.g. rigid parts. However, in the absence of prior knowledge on this structure, patches are preferably regularly distributed over the surface. To this purpose, we consider geodesic distances and derive an efficient greedy algorithm. The idea is to randomly choose a vertex to be the centre of the first patch and then to grow this patch until a maximum radius is reached. The subsequent patch centres are chosen among the unassigned vertices which lie on the most patch boundaries. The front of a new patch is propagated from the centre until a maximum radius is reached or until the processed vertex is closer to the centre of another patch. Applied on hexagonal lattices, this technique produces regularly sampled patches. Figure 2 illustrates the algorithm behaviour.

Based on this representation of the surface we model the pose of the patch P_i with a rotation matrix $\mathbf{R_i}$ and with the position of its center of mass $\mathbf{c_i}$. These parameters encode a rigid transformation with respect to the world coordinates. Given a vertex v on the mesh whose position on the reference pose was $\mathbf{x^0}$, and assuming v to move with the patch P_i, then its new position $\mathbf{x_i}(v)$ as predicted by P_i is :

$$\mathbf{x_i}(v) = \mathbf{R_i}(\mathbf{x}^0(v) - \mathbf{c_i^0}) + \mathbf{c_i}, \tag{1}$$

where $\mathbf{c_i^0}$ is the position of P_i's center of mass in the reference pose.

Since different patches yield different predictions for the point positions, the predicted coordinates are linearly blended to recover the deformed mesh. We found that a simple weighting scheme consisting in compactly supported Gaussians performed well. Each patch P_i defines a weighting function $\alpha_i(\nu) : \nu \mapsto \mathbb{R}$ whose support is the union of P_i and the adjacent patches. We used Gaussians of the Euclidean distance to the centre of mass of P_i as functions and normalized them so they would add up to 1 at every vertex. These $\{\alpha_i\}_{i \in patches}$ are only evaluated once on the reference mesh.

Rigidity Constraints Rigidity constraints can easily be expressed in this framework by enforcing the predictions of a point position by neighbouring patches to be consistent. We chose to use the the union of P_i and P_j as the region where two neighbouring patches should agree. This yields for each vertex $v \in P_i \cup P_j$ a rigidity term E_v^{ij} :

$$E_v^{ij} = (\alpha_i(v) + \alpha_j(v)) \, \|\mathbf{x}_i(v) - \mathbf{x}_j(v)\|^2. \tag{2}$$

It is worth of notice that we chose to use the blending basis functions $\{\alpha_i\}_{i \in patches}$ as weights in the rigidity term. Weighting the contribution of vertices in equation (2) allows a finer control on the local rigidity of the mesh. In practice we ran our experiments using the sum of the normalized blending basis functions. The total rigidity term is then simply defined as :

$$E_{rigidity} = \frac{1}{2} \sum_{P_i \in patches} \sum_{P_j \in N_i} \sum_{v \in P_i \cup P_j} E_v^{ij}, \tag{3}$$

where N_i is the set of patches neighbours to P_i.

3.2. Numerical Optimization

The mesh deformation can be expressed as the following optimization problem:

$$\underset{\{\mathbf{R_i}, \mathbf{c_i}\}_{i \in patches}}{argmin} \quad k_r E_{rigidity} + \sum f^2, \tag{4}$$

where k_r weights the regularization contribution and the f functions are the data terms that depend on the application.

A direct optimization of the above non-linear term involves rotations whose parametrization must be chosen with care. In contrast to [17, 10], we do not run the optimization explicitly on the parameters $\{\mathbf{R_i}, \mathbf{c_i}\}_{i \in patches}$. We follow instead [12, 1] that propose to use local perturbations of existing rotations with affine updates of the current rigid transformations as a numerically safer way to optimize with respect to rotations.

To simplify notations, $\mathbf{x_i}, \alpha_i$ denote the position and weight $\mathbf{x_i}(v), \alpha_i(v)$ of v respectively. The affine update is parametrized with a 6×1 vector $\boldsymbol{\omega_i} = (\mathbf{u_i}, \mathbf{v_i})$ and operates on the transformed vertex positions :

$$\mathbf{x_i} \mapsto \mathbf{x_i} + (\mathbf{x_i} - \mathbf{c_i}) \times \mathbf{u_i} + \mathbf{v_i}. \tag{5}$$

This parameterization is affine in $\boldsymbol{\omega_i}$ and can be rewritten as :

$$\begin{aligned} &\mathbf{x_i} \mapsto \mathbf{x_i} + \mathbf{K_i}(\mathbf{x_i})\, \boldsymbol{\omega_i} \\ &\text{where } \mathbf{K_i}(\mathbf{x_i}) = \left[[\mathbf{x_i} - \mathbf{c_i}]_\times \quad I \right]. \end{aligned} \tag{6}$$

Using the above parametrization, we can linearise both the regularization term and the data term up to first order as shown below. Thus the optimization of expression (4) becomes a quadratic problem for which we can derive an iterative scheme similar to Gauss Newton. The following paragraphs detail this scheme.

First-order approximations of the data term Any cost function f of vertices positions $\mathbf{x}$ can have its gradient wrt. $\boldsymbol{\omega_i}$ expressed using the chain rule and the fact that $\mathbf{K_i}(\mathbf{x_i})$ is precisely the Jacobian of $\mathbf{x_i}$ wrt. $\boldsymbol{\omega_i}$.

$$\left[\frac{\partial f}{\partial \boldsymbol{\omega_i}}\right] = \left[\frac{\partial f}{\partial \mathbf{x_i}}\right] \left[\frac{\partial \mathbf{x_i}}{\partial \boldsymbol{\omega_i}}\right] = \left[\frac{\partial f}{\partial \mathbf{x_i}}\right] \mathbf{K_i}(\mathbf{x_i}). \tag{7}$$

Therefore, the first order approximations of f yields the following quadratic form for f^2 :

$$f^2(\boldsymbol{\omega_i}) \simeq \| \left[\frac{\partial f}{\partial \boldsymbol{\omega_i}}\right] \boldsymbol{\omega_i} + f \|^2. \tag{8}$$

First-order approximations of the rigidity term The local rigidity energy E_v^{ij} described in equation (2) can also be rewritten using a first order development of the positions $\mathbf{x_i}, \mathbf{x_j}$ wrt. $\boldsymbol{\omega_i}, \boldsymbol{\omega_j}$ respectively as :

$$\begin{aligned} E_v^{ij}(\boldsymbol{\omega_i}) \simeq (\alpha_i + \alpha_j)\, \| &(\mathbf{x_i} + \mathbf{K_i}(\mathbf{x_i})\, \boldsymbol{\omega_i}) \\ &- (\mathbf{x_j} + \mathbf{K_j}(\mathbf{x_j})\, \boldsymbol{\omega_j}) \|^2. \end{aligned} \tag{9}$$

Grouping all the $\boldsymbol{\omega_i}$'s in a $6 \times N_{patches}$ vector $\boldsymbol{\omega}$ and using matrix notations, the local quadratic approximation of $E_{rigidity}$ becomes :

$$E_{rigidity}(\boldsymbol{\omega}) \simeq \|\mathbf{WG}\boldsymbol{\omega} - \mathbf{Wb}\|^2, \tag{10}$$

where the $\mathbf{K_i}(\mathbf{x_i})$ and $-\mathbf{K_j}(\mathbf{x_j})$ are stacked on the lines of a matrix $\mathbf{G}$, their respective weights $\alpha_i + \alpha_j$ in a matrix $\mathbf{W}$ and their right hand side $\mathbf{x_j} - \mathbf{x_i}$ in a vector vector $\mathbf{b}$.

Finding an energy-decreasing step Finally expressions (8) and (10) lead to the following quadratic approximation of equation (4) :

$$\underset{\boldsymbol{\omega}}{argmin} \left\| \begin{bmatrix} k_r\, \mathbf{WG} \\ \left[\frac{\partial f_1}{\partial \boldsymbol{\omega}}\right] \\ \vdots \end{bmatrix} \boldsymbol{\omega} - \begin{bmatrix} k_r\, \mathbf{Wb} \\ f_1 \\ \vdots \end{bmatrix} \right\|^2. \tag{11}$$

Solving the above least-squares problem yields a step $\boldsymbol{\omega}$. However the local affine updates from equation (6) are not rigid updates of the transformations as they induce scaling effects. In [1, 2] it is suggested that the closest rigid transformation can be efficiently recovered using the Horn algorithm [9]. For each patch P_i and a given update $\boldsymbol{\omega_i}$, the Horn algorithm is applied between the $\mathbf{x_0}$ and the corresponding $\mathbf{x_i} + \mathbf{K_i}(\mathbf{x_i})\, \boldsymbol{\omega_i}$ to get an updated approximation of $(\mathbf{R_i}, \mathbf{c_i})$.

Note that importantly, once we have recovered the updated $\{\mathbf{R_i}, \mathbf{c_i}\}_{i \in patches}$ from $\boldsymbol{\omega}$ and computed the resulting interpolated mesh, there is no guarantee that the original energy of equation (4) decreases. We thus scale the $\boldsymbol{\omega}$ vector so that the corresponding updated $\{\mathbf{R_i}, \mathbf{c_i}\}_{i \in patches}$ decreases the energy. Finally, we iteratively solve (11) and recover rigid transformations from $\boldsymbol{\omega}$ until convergence.

Remarks In practice, the $\mathbf{G}$ matrix is never explicitly manipulated to solve (11) but instead it is easier to compute $6N_{patches} \times 6N_{patches}$ matrix $\mathbf{G^T W^T WG}$. This matrix is very sparse except for 6×6 blocks and is a Laplacian matrix on the graph of patches. It is rank-deficient. If the mesh has only one connected component, its kernel is of dimension 6 which corresponds to the 6 degrees of freedom of one patch. This means that the row space of the gradients of the data functions must be at least of dimension 6.

4. Applications and Experiments

4.1. Monocular Cloth Tracking

Recovering the evolutions of 3D surfaces from 2D information is a highly under-constrained problem. Prior knowledge is therefore required to ensure consistent deformations. In this section we show that our framework, equipped with simple surface rigidity priors, performs well in this situation. We use the data made available by the EPFL computer vision group for that purpose [13]. It consists of a reference 3D mesh model of a piece of cloth and of a list of correspondences for each video frame. Each of these correspondences maps a 3D position on the reference mesh, expressed in barycentric coordinates, to a 2D position in the image. We use these correspondences in our framework by simply defining for each of them two data functions f_u and f_v that measure the reprojection error of a 3D vertex on each image axis.

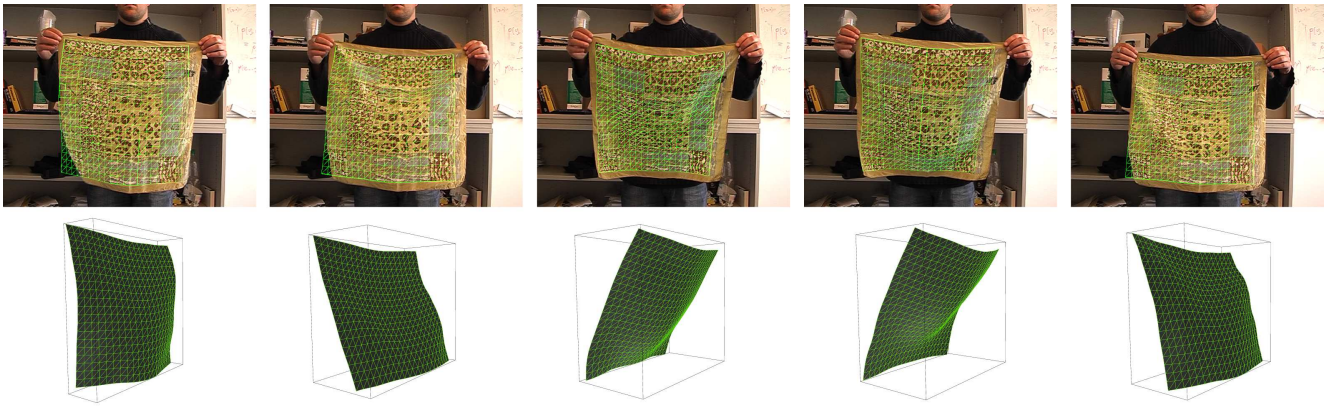
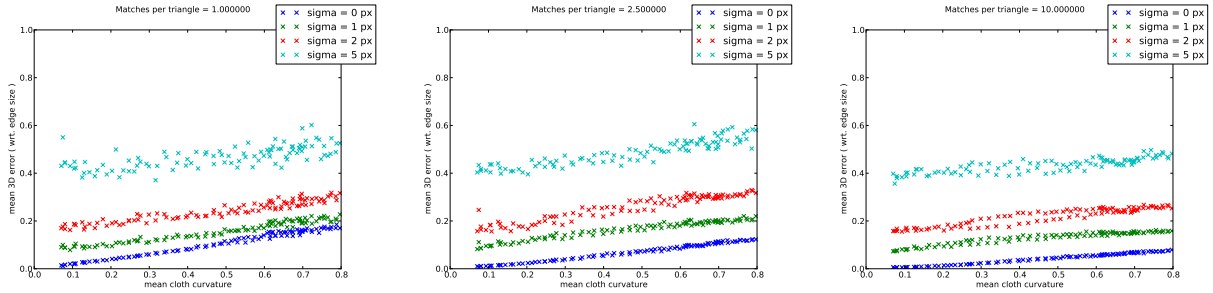


Figure 3. Results on the cloth dataset [13]. On the top row, the mesh overlay on the original data demonstrates a low reprojection error, while the bottom row shows that the recovered 3D deformations are physically plausible. Note on the two first images that in absence of matches, the rigidity constraints make the mesh return locally to its rest pose (flat cloth).



(a) 1 match per triangle (b) 2.5 matches per triangle (c) 10 matches per triangle

Figure 4. Results on synthetic monocular data: we present the 3D error of the recovered deformation for different levels of noise on the 2D matches. As in [13] we plot this error as a function of the average curvature over the ground truth mesh. Here the curvature is computed by averaging the angles between adjacent triangles.

Qualitative Evaluation: the cloth dataset Figure 3 presents overlays of the tracked mesh over the input images. They demonstrate low residual reprojection errors. The figure also shows side views of the corresponding 3D shape illustrating the coherence of the recovered 3D deformations. On this dataset, the computation time was approximately of 5 frames per second.

Quantitative evaluation : resilience to noise This experiment is based on motion capture data of the deformation of a real piece of cloth. In a way similar to [13] we synthesized 2D-3D correspondences and added centered 2D-Gaussian noises to evaluate the behavior of our framework in the presence of noise. Figure 4 shows the results of the quantitative evaluation. We evaluated our algorithm using 1, 2.5 and 10 matches per triangle on the reference mesh, composed of 96 triangles. The presented error is the average 3D distance of the recovered vertices to their ground truth positions. Following [13], we do not plot the temporal evolution of the error but sort the frames using the average curvature over the ground truth mesh and plot the error in respect to it.

4.2. Multi-camera environment

A major application of mesh tracking methods concerns multi-camera environments. In this context, our framework deforms a reference model across time so that it fits independently reconstructed sets of points and normals that were obtained using multi-view reconstructions. The procedure is similar to the ICP algorithm as it constantly re-evaluates point correspondences between the template and the target point clouds.

We ran our experiments on several standard datasets in the field (see Figures 5, 6 and 7). Interestingly, [18, 7] tackle these sequences using strong assumptions on the articulated nature of the mesh in the form of a skeletal model. In contrast, and closer to our framework, [5] assumes only the volumetric nature of the tracked object. Our method goes fur-

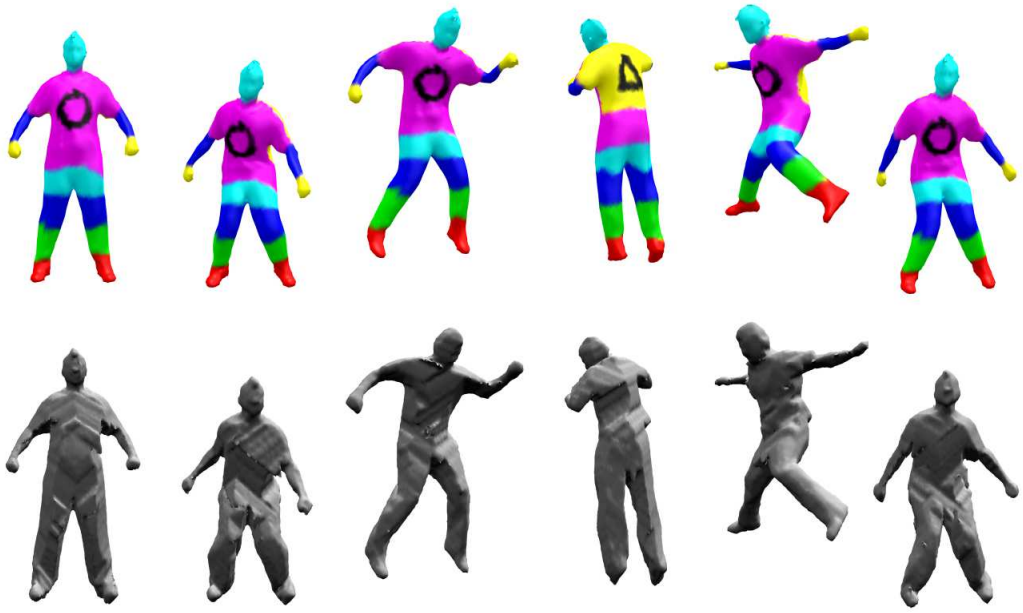


Figure 5. Results on the MIT bounce sequence [18]. A simplified version of the provided template mesh is used as reference. It is deformed across time (top row) to fit the points and normals of visual hull models (bottom row).

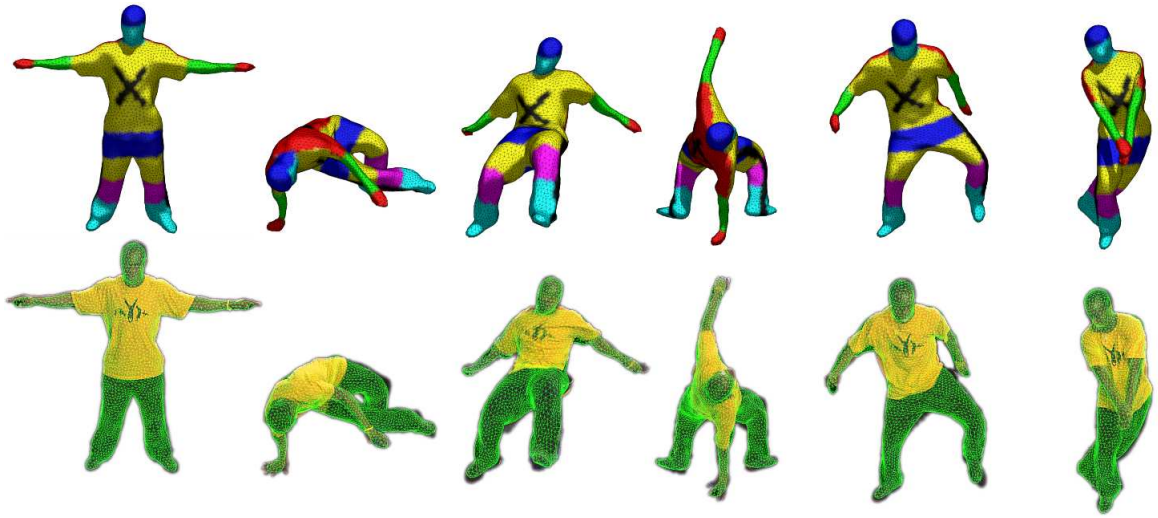


Figure 6. On the Free sequence from the SurfCap project [16], the first reconstruction is used as reference mesh and deformed to fit the points and normals of independently reconstructed photoconsistent mesh models. We show on the bottom row the overlay of the resulting mesh deformation on the original image data.

ther and is purely surface-based. Note that similarly to [18], we only use geometric data and do not consider geodesic or photometric sparse feature matches.

Data Term The energy term that is to be minimized at each step is the sum of the distances from the target geometry points to the deformed template :

$$f_{ICP}^2 = \sum_{\mathbf{x_t} \in TargetCloud} \underset{v \in \nu}{min} \|\mathbf{x}(v) - \mathbf{x_t}\|^2. \qquad (12)$$

As in the original ICP algorithm, we compute dense associations between the target geometry and the current deformation of the template. However, false associations affect the minimization more than in a rigid ICP, due to the many

degrees of freedom of the deformation. To regularize these correspondences we use the Horn algorithm [9] to compute for each patch P_i an average rigid motion mapping its vertices in the reference pose to their targets. This computed rigid transformation defines a function $\mathbf{x_i^{ICP}}$ that applies to the patch P_i and its neighborhood $P_j \in N_i$. Residual errors on each of them are computed as the average distance between the predicted patches and the target surface. This yields weights $w_{i|i}$ and $\{w_{i|j}\}_{P_j \in N_i}$. We then introduce for each vertex $v \in P_i$ a sum of quadratic terms to be minimized :

$$f_{ICP}^2 \simeq \sum_{P_i \in patches} \sum_{v \in P_i} \Big(w_{i|i} \|\mathbf{x}(v) - \mathbf{x_i^{ICP}}(v)\|^2 + \sum_{j \in N_i} w_{i|j} \|\mathbf{x}(v) - \mathbf{x_j^{ICP}}(v)\|^2 \Big). \quad (13)$$

These terms yield easily computable Jacobians wrt. $\boldsymbol{\omega}$ using the equation (7) and fit exactly our framework as elements of the local quadratic approximation of the cost function.

In practice, the optimization loop operates at three different levels of resolution, in a coarse-to-fine manner, starting with large averaging effects for point correspondences to increase the convergence speed. The first two levels use the result of the previous frame as the reference pose for rigidity priors. This allows to preserve small scale deformations before the third and last optimization. This last high resolution optimization is usually initialized close to the solution and uses the first mesh of the sequence as rigidity prior, which prevents long term drifting in the rigidity.

Results We present results on different type of sequences. Figure 5 shows the behavior of our algorithm when confronted to noisy voxel carving reconstructions. Figure 6 illustrates a long sequence with complex deformations and fast motion. These figures show that on standard sequences our algorithm provides qualitatively comparable results to state of the art methods, while requiring much less knowledge on the object of interest. Figure 7 shows our results on a scene involving three distinct objects. Such complex scenes are very challenging for state of the art approaches and have to the best of our knowledge not been addressed until now.

5. Discussion and future work

Although its versatility was demonstrated in the previous section, the presented framework has limitations. Firstly, representing surface deformations using patches and assuming each one transforms rigidly increases the robustness. However, it limits the ability to recover curvature changes at scales smaller that the patch radius. Secondly, using a reference mesh constrains the topological nature of the object that can not evolve over the sequence. In the results shown in Figure 7 for example, the reference mesh (top line) is topologically suitable in that the three objects are three distinct connected components.

6. Conclusion

We presented a mesh deformation framework which successfully deals with noisy data coming form different sources, large deformations and fast motion. It assumes little knowledge on the nature of the observed surfaces and uses a simple strategy to divide the original shape into regularly sampled regions called patches. The motion constraints inferred from the data terms are averaged on every patch, while a rigidity constraint between adjacent patches ensures physically plausible deformations. We demonstrated the versatility and efficiency of our approach by evaluating it on two different applications: monocular cloth tracking and surface tracking in multi-camera environments. The flexibility of our method is especially outlined by the results on the ball sequence where three distinct objects interact. To the best of our knowledge surface tracking in this kind of complex scene has not been addressed by previous works.

Acknowledgments

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

References

[1] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. Primo: Coupled prisms for intuitive surface modeling. *Eurographics Symposium on Geometry Processing*, 2006.

[2] M. Botsch, M. Pauly, M. Wicke, and M. H. Gross. Adaptive space deformations based on rigid cells. *Comput. Graph. Forum*, 2007.

[3] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 2008.

[4] J. Chai, J. Xiao, and J. K. Hodgins. Vision-based control of 3d facial animation. In *In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003.

[5] 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.*, 2008.

[6] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. In *CVPR*, 2008.

[7] 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 *CVPR*, 2009.

[8] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the ACM SIG-*

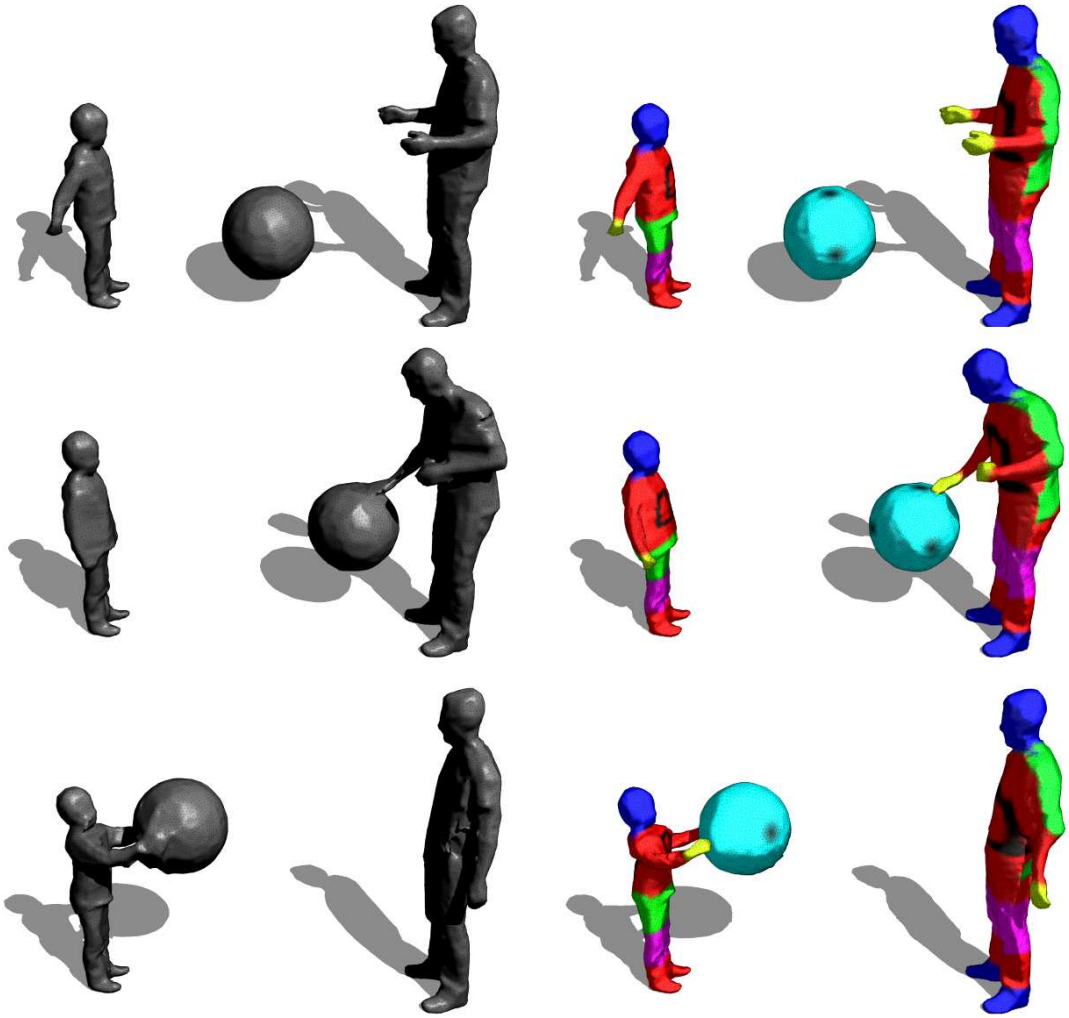


Figure 7. Results on the multi-view ball dataset from the INRIA-Perception. The left column shows the independently reconstructed photoconsistent models. The right column shows the deformations of the reference mesh obtained by fitting the sequence of photoconsistent models. Note that on the middle line, the arm is correctly tracked while it is no present in the observation.

GRAPH/Eurographics symposium on Computer animation, 2003.

[9] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 1987.

[10] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Comput. Graph. Forum*, 2008.

[11] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *ICCV*, 2009.

[12] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3d shapes. *IJCV*, 2006.

[13] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3d surface registration. In *ECCV*, 2008.

[14] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for nonrigid 3d shape recovery. *PAMI*, 2007.

[15] O. Sorkine, D. C. Or, Y. Lipman, M. Alexa, C. Rössl, and H. P. Seidel. Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, 2004.

[16] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 2007.

[17] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*. ACM, 2007.

[18] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*. ACM, 2008.