Temporal Surface Tracking Using Mesh Evolution

Kiran Varanasi, Andrei Zaharescu, Edmond Boyer, Radu Horaud

Temporal Surface Tracking using Mesh Evolution

Kiran Varanasi, Andrei Zaharescu, Edmond Boyer, Radu Horaud

LJK - INRIA Rhône-Alpes, France

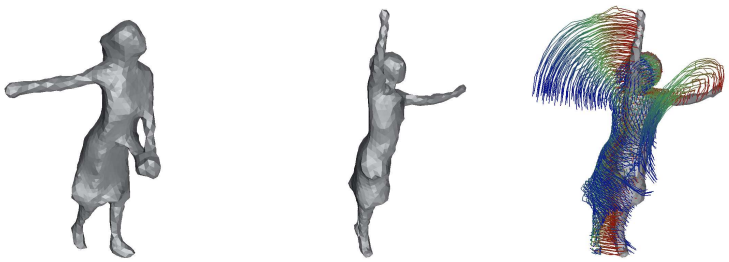
Abstract. In this paper, we address the problem of surface tracking in multiple camera environments and over time sequences. In order to fully track a surface undergoing significant deformations, we cast the problem as a mesh evolution over time. Such an evolution is driven by 3D displacement fields estimated between meshes recovered independently at different time frames. Geometric and photometric information is used to identify a robust set of matching vertices. This provides a sparse displacement field that is densified over the mesh by Laplacian diffusion. In contrast to existing approaches that evolve meshes, we do not assume a known model or a fixed topology. The contribution is a novel mesh evolution based framework that allows to fully track, over long sequences, an unknown surface encountering deformations, including topological changes. Results on very challenging and publicly available image based 3D mesh sequences demonstrate the ability of our framework to efficiently recover surface motions .

1 Introduction

Tracking the surface of moving objects is of central importance when modeling dynamic scenes using multiple videos. This key step in the modeling pipeline yields temporal correspondences which are necessary when considering motion related applications such as motion capture. Furthermore, it allows recovery of improved and consistent descriptions of object shapes and appearances.

In this work we address the problem of capturing the evolution of a moving and deforming surface, in particular moving human bodies, given multiple videos. A large variety of directions can be followed, depending on the *a priori* knowledge of the observed shape, on the representation chosen for surfaces and on the information taken into account for deformations. *Model-based approaches* assume a known model of the observed surface, which is tracked over time sequences, hence solving for time correspondences. This model can be locally rigid, e.g [1,2,3], or deformable, e.g. [4,5]. Unfortunately, exact models need to be available, which is seldom the case in general situations. In particular, the topology of the surface can evolve over time as shown in Figure 1. As a consequence, approaches in this category are restricted to specific scenarios.

In contrast, non model-based approaches try to find displacement fields between 2 different instants in the sequence. In this category, *scene flow approaches* consider dense vector fields with various representations including voxels [6,7], implicit representations [8] or meshes [9]. However, the associated differential methods are limited to small displacements between successive frames. Alternatively, *feature-based approaches* [10,11,12] consider meshes and allow for larger motions by casting the



(a) (b) (c)

Fig. 1. An example of a surface for which the topology can hardly be known in advance. The belt of the dress forms a new protrusion that appears and disappears (a)-(b). (c) Dense point trajectories computed from (a) to (b)

problem as a labeling between 2 meshes using local geometric or photometric information. This labeling solves for partial correspondences between 2 frames only and might lose efficiency when applied over long sequences, in particular as topological changes occur. Our objective is different but complementary and targeted at providing full mesh evolution over time sequences.

Our approach is grounded on the observation that natural surfaces are usually arbitrary shaped and difficult to model *a priori*. In addition, shapes can significantly evolve over a time sequence. For instance, human bodies are usually covered by clothes whose topologies can change. To handle such deformations, we use meshes which are morphed from one frame to another. Like feature-based approaches, we use photometric cues provided by images and geometric cues provided by the recovered meshes. However, instead of looking for a dense match between the vertices of the 2 meshes, we use a sparse, but robust set of matches and its associated displacement vector field to drive a full consistent mesh evolution, with possible topological changes. This approach provides both a consistent surface evolution over time and dense point trajectories on the surface.

The framework we propose assumes little about the observed surface, thus relaxing the constraints for markers, known models or limited deformations and displacements. It allows for the recovery of trajectories of points, as shown in Figure 1, on a surface undergoing significant deformations including topological changes. Instead of using traditional *Eulerian methods* e.g. level sets [13], a major innovation is to cast the problem within a mesh evolution framework that performs mesh morphing, thereby avoiding *Eulerian* limitations such as complexity and inappropriateness for tracking interface properties, e.g. vector displacements.

The remainder of this paper is organized as follows. Related works are reviewed in section 2. The proposed approach is outlined in section 3. The recovery of displacement vector fields is described in section 4 and 5. The mesh deformation is then explained in section 6. Experimental results obtained with publicly available sequences are shown in section 7, before concluding in section 8.

2 Related Work

Surfaces observed in multiple views can be fully tracked through a deformable model, which is fitted to image related observations, see for instance [14] for a review on 3D deformable models. This operation appears to be difficult, unless a precise model is available. This is particularly true with recent works [9,15] that propose to use a laser-scanned model of the surface prior to tracking. Unfortunately, precise models will not be available in general situation. Moreover the fixed topology assumption significantly limits the application domain.

An alternative is to directly estimate surface motions between temporal frames and a significant effort has been put in that direction over the past years. *Scene flow approaches* recover dense motion fields using derivatives of the image signal [6,7,16]. In [8] this is used within a variational framework to fully track surfaces using level sets. As noticed in [12], flow-based approaches are nevertheless limited to small displacements, as a consequence of finite difference approximations of derivatives.

Another class of approaches solve for shape matching. Assume that shape models, e.g. meshes, can be recovered from images independently over time sequences, using for instance [17,18,8]. Then temporal correspondences can be obtained through vertex mapping between successive meshes. While providing displacement fields between frames, temporal correspondences yet only partially solve the problem of surface tracking since the transformation that maps a surface onto another remains unknown. Nevertheless, this can be seen as a first step towards full surface tracking. The associated labeling problem can be solved in various ways. Point based approaches, e.g. [19,20], register sets of points but do not account for shape information, i.e. mesh connectivity. More closely related to our framework, numerous mesh based approaches have been proposed. Some solve for correspondences indirectly through embeddings, e.g. [21,22,23], with the price of an often difficult intermediate step. Other approaches bypass this step and directly seek correspondences between meshes. For instance [10,11] successfully match 2 different poses of a *proper* mesh, e.g. a range-scanned model, using geometric features only. However, these approaches do not easily extend to real objects' surfaces recovered from images since the associated meshes can vary drastically between successive frames and furthermore, their topologies can change. In that case, photometric cues are advantageously added to geometric features to make the labeling feasible as in [12]. While allowing for large motions, labeling approaches seek dense correspondences, which are difficult to obtain on a regular basis over long sequences.

To the best of our knowledge, no previous work has attempted to perform the full tracking of an unknown mesh undergoing deformations with possibly topological changes, using multiple videos. Our method bridges the gap between model-based and non model-based approaches since we evolve a mesh using temporal correspondences. To this purpose, we build on some of the ideas already used in the approaches mentioned above. We combine the interest of both photometric and geometric cues for robust matching [12] with Laplacian diffusion [9] to get a dense displacement field. The resulting vector field is used to initialize a consistent mesh evolution between successive frames. To demonstrate the robustness of our scheme, we have used challenging real data sets with large motion and topological changes.

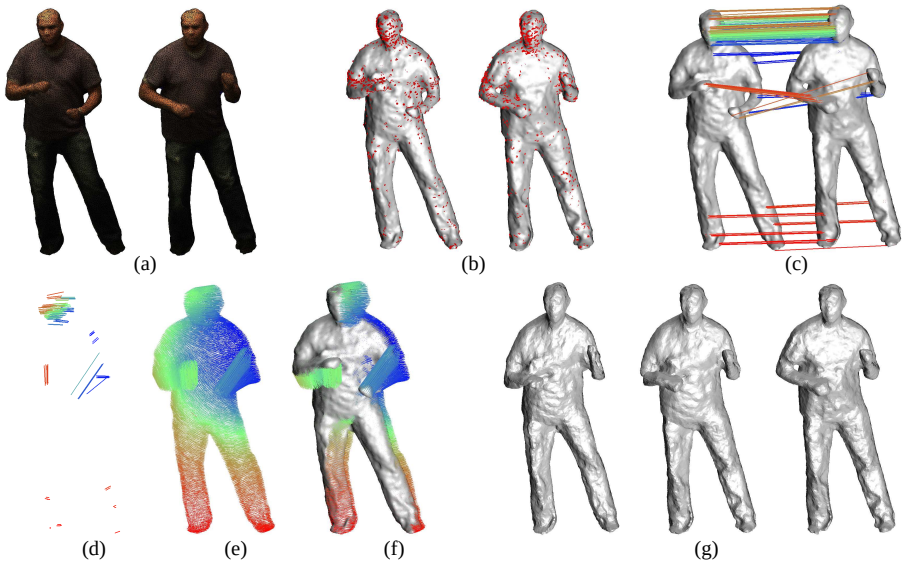


Fig. 2. The different consecutive steps of the proposed framework: (a) original meshes $\mathcal{S}^t$ and $\mathcal{M}^{t+1}$; (b) features extracted; (c) feature matching; (d) the associated sparse-displacement; (e)-(f) the dense displacement field after Laplacian diffusion; (g) mesh-morphing (observe the topological change that takes place around the right arm of the model - the right elbow de-attaches from the body, creating a genus change).

3 Approach Outline

We consider multiple camera environments and we assume that multiple calibrated videos of an object with closed surfaces are available. We also assume that 3D mesh models $\mathcal{M}^{t\in[1..n]}$ of the object at different time instances $[1..n]$ estimated using multi-view 3D modeling approaches, e.g. [17,18,8], are available. These meshes $\mathcal{M}^{t\in[1..n]}$ correspond to discrete values of the time continuous mesh $\mathcal{S}^t$. In order to recover $\mathcal{S}^t$, the mapping of $\mathcal{S}^t$ onto $\mathcal{M}^{t+1}$ is iteratively estimated using the following 3 consecutive steps, starting with $\mathcal{S}^1 = \mathcal{M}^1$:

1. Sparse match: photometric and geometric cues are used to match a set of points between $\mathcal{S}^t$ and $\mathcal{M}^{t+1}$ (see Figure 2-c). Unlike previous approaches, only a sparse set of correspondences is expected.
2. Motion diffusion: the identified correspondences define a sparse displacement field over $\mathcal{S}^t$ (cf. figure 2-d). This field is propagated over all vertices by Laplacian diffusion hence preserving local shape details [24] (cf. Figure 2-e).
3. Mesh evolution: The dense displacement field is applied to the vertices of $\mathcal{S}^t$ yielding a new mesh. The resulting mesh $\hat{\mathcal{S}}^t$ is then morphed to $\mathcal{M}^{t+1}$ by minimizing the signed distance to $\mathcal{M}^{t+1}$ (cf. Figure 2-fg). Mesh consistency within the optimization is enforced using [25]. The final optimized mesh defines $\mathcal{S}^{t+1}$.

The Laplacian diffusion allows a partial vertex matching only and yields to a good estimation of the motion at all vertex locations. Nevertheless, an additional step is required to guarantee that the resulting mesh fits the observations $\mathcal{M}^{t\in[1..n]}$ and also to guaranty its correctness, e.g. manifoldness. This is in contrast with the work [9] which also uses Laplacian diffusion to evolve a reference mesh to the observed posture, but without refinement and therefore without guaranties. The 3 above steps are detailed in the following sections.

4 Feature Matching

The primary step of our approach is to obtain a set of good feature matches across the two frames. In contrast to the labeling approaches mentioned previously, we do not intend here to produce a dense match over mesh vertices, but only a robust selection. To this end, we first detect a set of interest points and provide them with a variety of distinctive features (photometric and geometric). These sets of feature vectors are then matched across in an exhaustive manner, to compute a preliminary set of potential matches. The error of the matching is defined in terms of difference between the different feature vectors. We employ a two-step minimization procedure (a coarse step will guide a finer step) in order to avoid local minima. We detail each of these steps in the following subsections.

4.1 Feature Extraction

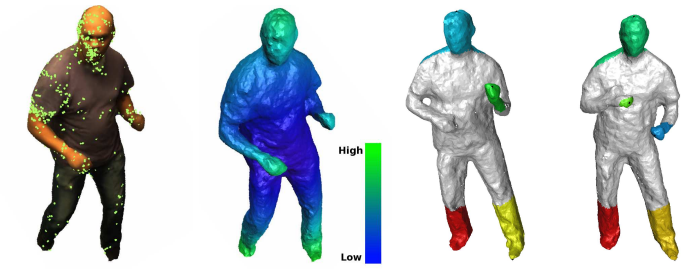
For each frame, we are provided with a 3D mesh representation S^t coupled with a set of images I_i^t depicting camera views of the object from different angles.

Image Features. We use corners as image features. If silhouettes are available, we use them to constrain the features (in practice, we erode the silhouettes by $\alpha = 3$ pixels to eliminate the features close to the boundary). The feature points are computed as maxima of the determinant of the image Hessian matrix. We have chosen Speeded-Up Robust Features (SURF) [26] as an image descriptors, because of their robustness in wide-baseline stereo and because of their increased speed of computation due to integral images. We back-project the detected interest points onto the 3D mesh and assign the corresponding SURF feature, together with color features, i.e. hue, saturation and value (HSV). The color for the 3-D point is calculated as the median color in the visible cameras. Figure 3-a illustrates the distribution of the feature points over a sample 3D mesh.

Mesh Features. Geodesic distances between mesh points offer crucial information in matching non rigid shapes. We use a feature called the *normalized geodesic integral* [27], which is defined as:

$$\mu(V) = \int_{P\in S} G(V,P)dS \quad , \quad \mu_n(V) = \frac{\mu(V) - Min_{P\in S}\mu(P)}{Max_{P\in S}\mu(P)}$$

where $G(V, P)$ denotes the geodesic distance between the points P and V and $\mu(V)$ is defined as the sum of the geodesic distances from V to all points on S. After normalization, $\mu_n(V)$ provides a continuous function whose value indicates the apparent nearness of a point to the center of the object. Its maxima will correspond to the extremities of the object. Figure 3-b illustrates the distribution of this function over the sample 3D mesh.



(a) (b) (c)

Fig. 3. (a) SURF feature points back-projected onto the 3D mesh (b) The geodesic integral function over the 3D mesh (c) Surface Protrusions detected as the maxima of the geodesic integral - the left figure has a collapsed protrusion

4.2 Coarse Matching by Surface Protrusions

We proceed to identify the extrema of the geodesic integral $\mu_n(v)$. This is done by simply imposing a threshold on the value of $\mu_n(v)$ and selecting the points on the mesh that lie above this threshold. Such points lie in compact clusters, typically corresponding to the different protrusions of the object. As shown in Figure 3-c, between time-frames, some of these protrusions collapse onto the surface inducing changes in the topology of the mesh. We will devise an algorithm whose goal is to correctly detect the topological changes and match the extrema accordingly.

We select the local extremum of the function $\mu_n(v)$ as representative for each cluster. The extent of each protrusion is defined by a local geodesic neighborhood from the representative point. We assign a feature to the protrusion based on color distribution in this region. This is a highly distinctive feature defined on a large neighborhood. These features can be visualized as in Figure (4- a). We now proceed to match each of these protrusions uniquely across the two frames. This problem is formulated as an error minimization problem with the following error:

$$E = \sum_i \Psi(X_i^t, X_i^{t+1}) + \sum_{i,j} |G(X_i^t, X_j^t) - G(X_i^{t+1}, X_j^{t+1})|$$

where X_i^{t+1} denotes the match of a protrusion X_i^t, $\Psi(X_i^t, X_i^{t+1})$ denotes the error computed through color features and $G(X_i, X_j)$ denotes the geodesic distance between

(a) (b) (c)

Fig. 4. (a) Color features computed on surface protrusions (b) The surface clustered into regions based on color (c) Surface protrusions matched with each other. Note that even the collapsed protrusion is matched correctly.

the protrusions X_i and X_j. Since the number of detected protrusions is typically very small, we proceed to do an exhaustive search to solve the matching problem.

In the particular case of tracking adjacent frames, we add an additional term $||X_i^t - X_i^{t+1}||$ in the error, which denotes the Euclidean distance between the two matched protrusions. This means that the two matched protrusions are not too far from each other - a strong but valid assumption in our case. This helps the algorithm in resolving issues raised by symmetry (such as between the two hands, or between the two legs).

For a human body in a normal condition, the number of protrusions are 5 (head + 2 hands + 2 legs). When there is a collapse (hand/leg touching a part of the body), the number of detected protrusions will be 4 or lesser. There will be an implicit mismatch in the number of protrusions detected, and this will be detected trivially. The correspondences of such collapsed protrusions will be left without a match in the previous minimization step. It should be noted that such collapses will not damage the geodesic distances between protrusions which did not collapse themselves. In this way, we efficiently use the geodesic distances only where they are meaningful.

To handle collapsed protrusions, we cluster the target surface region based on color, and select the most appropriate cluster based on the error $\Psi(X_i^t, X_i^{t+1})$ (or $\Psi(X_i^t, X_i^{t+1}) + ||X_i^t - X_i^{t+1}||$) for adjacent frames. The surface regions clustered according to color are shown in Figure 4-b. The result of the matching is shown in Figure 4-c.

4.3 Fine Matching by Feature Points

The previous step provides a good initialization to perform feature matching at a finer level. We intend to produce a selection of feature matches that are representative of the surface and that are mutually consistent with each other. We define an error function based on color, SURF features and the array of geodesic distances of the feature points from each of the protrusions which are matched successfully in the above step (without collapse). We select a set of best features P_i^t, P_i^{t+1} based on this error, and prune this further to impose mutual spatial consistency.

If two feature points P_1^t, P_2^t are found to be geodesically near to each other, we connect them by a link which encodes the Euclidean length $|\overrightarrow{P_1^t P_2^t}|$ of the line joining

the two points, and the angles it makes with the normals $\hat{P}_1^t, \hat{P}_2^t$ at both the ends. Then the two pairs of matches (P_1^t, P_1^{t+1}) and (P_2, P_2^{t+1}) are checked for mutual spatial consistency in terms of the elastic stretch (γ_s) and twist (γ_{t1}, γ_{t2}) of the link, defined as:

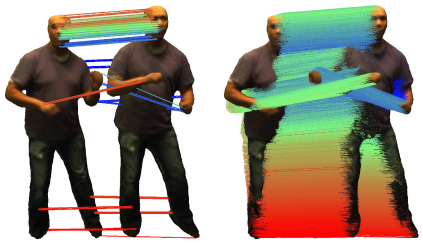
$$\begin{aligned}\gamma_s &= \Gamma_s(|\overrightarrow{P_1^t P_2^t}| - |\overrightarrow{P_1^{t+1} P_2^{t+1}}|)\\ \gamma_{t1} &= \Gamma_t(\theta(\overrightarrow{P_1^t P_2^t}, \hat{P_1^t}) - \theta(\overrightarrow{P_1^{t+1} P_2^{t+1}}, \hat{P_1^{t+1}}))\\ \gamma_{t2} &= \Gamma_t(\theta(\overrightarrow{P_1^t P_2^t}, \hat{P_2^t}) - \theta(\overrightarrow{P_1^{t+1} P_2^{t+1}}, \hat{P_2^{t+1}}))\end{aligned}$$

where $\theta(\boldsymbol{v1}, \boldsymbol{v2})$ denotes the angle between two vectors, and Γ_s, Γ_t denote two Gaussian penalty functions.

Furthermore, pairs are checked for parity in order (ρ_1) and orientation (ρ_2) with respect to the nearest matched protrusion (X^t, X^{t+1}), and defined as:

$$\begin{aligned}&\rho_1 : Sign(|\overrightarrow{P_1^t X^t}| - |\overrightarrow{P_2^t X^t}|) = Sign(|\overrightarrow{P_1^{t+1} X^{t+1}}| - |\overrightarrow{P_2^{t+1} X^{t+1}}|)\\ &\rho_2 : \theta(\overrightarrow{P_1^t X^t} \times \overrightarrow{P_2^t X^t}, \overrightarrow{P_1^{t+1} X^{t+1}} \times \overrightarrow{P_2^{t+1} X^{t+1}}) < 180^\circ\end{aligned}$$

An example of the set of identified feature matches is shown in Figure 5-a.



(a) (b)

Fig. 5. (a) The set of identified feature matches (b) The dense motion field computed by Laplacian diffusion.

5 Motion Diffusion

The feature matches computed as above provide the initialization for the transformation of the mesh. We propagate them across the entire mesh by Laplacian diffusion.

5.1 Approach

The geometric Laplacian operator is a way of encoding the local curvature of the mesh. This has proven to be useful in a variety of mesh applications [24], such as interactive mesh editing. This operator provides an efficient approach to deform the mesh while

preserving the local shape information. If N_V is the number of vertices, the Laplacian matrix L of size $N_V \times N_V$ is defined by these equations

$$\begin{aligned} L(i,j) &= wt(i,j) \quad \forall_{j \in N(i)} \\ L(i,j) &= 0 \quad \forall_{j \notin N(i)} \\ L(i,i) &= -1 * \textstyle\sum_{j \in N(i)} wt(i,j) \end{aligned}$$

where $N(i)$ is the set of vertices sharing an edge with the vertex i, and $wt(i,j)$ is the weight of the edge as defined by mean-value coordinates [24]. We compute the differential coordinates of the mesh at time t into three vectors $\delta X^t, \delta Y^t$ and δZ^t, where $\delta X^t = L * X^t$ (similarly for δY^t, δZ^t).

The feature matches computed earlier as initialization, we now define 3 matrices L_x, L_y and L_z, corresponding to the three dimensions X, Y and Z. If the number of feature matches is N_F, these matrices shall be of order $(N_V + N_F) \times N_V$. The first N_V rows shall be identical to the L matrix. The later rows are defined by constraints ($\forall i \in \{features\}$) (similarly for L_y, L_z):

$$\begin{aligned} L_x(i,j) &= 0 \quad \forall j \neq i \\ L_x(i,i) &= \lambda \end{aligned}$$

where λ is a weighting factor we set to 4000.

Similar to the matrix L_x, we append the vector δX^t by adding N_F new elements $\{\lambda * X_F^{t+1}\}$ where X_F^{t+1} are the X-coordinates of the feature matches in the frame $t+1$. The diffusion of the matches is done as a matrix inversion.

$$X^{t+1} = (L_x^\top L_x)^{-1} L_x^\top * \delta X^t$$

The matrix L_x being extremely sparse, this inversion can be efficiently implemented using Cholesky factorization.

Thus we propagate the mesh S^t via Laplacian diffusion to $\hat{\mathcal{S}}^t$. An example of the dense motion field obtained from a sparse set of feature matches is shown in Figure 5-b. In [9] the Laplacian operator is also used to diffuse motion information over meshes. However, motion is limited to rotation since flow information is considered. In contrast, we propagate full displacement vectors as obtained by matching feature points between $\hat{\mathcal{S}}^t$ and the observed mesh $\mathcal{M}^{t+1}$.

6 Mesh Deformation

The matching and diffusion steps presented in the previous sections provide us with a dense displacement field over the mesh $\mathcal{S}^t$. As mentioned before, such motion field is a good estimate of the true motion field between time t and $t+1$. However, it will not guarantee the exact overlap with the mesh observed at $t+1$, i.e. $\mathcal{M}^{t+1}$, nor the correctness of the resulting mesh. Therefore, a final step is needed in order to ensure both convergence to the observations and correctness. Our approach is motivated by the fact that the solution mesh $\mathcal{M}^{t+1}$ and the propagated Laplacian mesh $\hat{\mathcal{S}}^t$ are different, but nearby. Thus a solution is to perform surface-morphing, that is starting from the source surface, i.e. $\hat{\mathcal{S}}^t$, and evolving it towards the destination surface $\mathcal{M}^{t+1}$. To

this purpose, we have used [25] as an explicit surface evolution approach which handles self-intersections and topological changes and guarantees correctness. To drive the surface evolution, we adopt here a simple morphing scheme introduced in [28] and described below.

6.1 Approach

Consider an open set $O_A \subset \mathbb{R}^3$ representing the source object, enclosed by surface $S_A = \partial O_A$, and similarly the open set $O_B \subset \mathbb{R}^3$ representing the target object, enclosed by $S_B = \partial O_B$. Consider the signed distance u_B of S_B, as defined by:

$$u_B(x) = \begin{cases} -d(x, S_B) & \forall x \subset O_B, \\ d(x, S_B) & \text{otherwise}, \end{cases} \tag{1}$$

where $d(x, y)$ is the Euclidean distance between x and y in $\mathbb{R}^3$. Following [28], the surface motion that maximizes the overlap between the morphed object and S_B is defined by:

$$\frac{\partial S}{\partial t} = -u_B(x)\mathbf{N}(x), \tag{2}$$

where x is a point on the surface S and $\mathbf{N}(x)$ is the normal to S at x. The strategy described above will converge to the desired solution if the surface of departure S_A and the destination surface S_B overlap.

6.2 Discussion

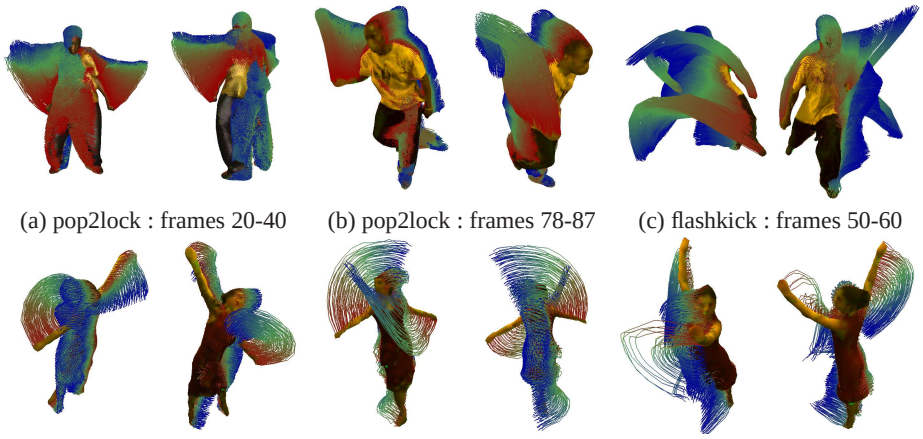
In a few cases, certain tracks are temporarily lost, due to the non-overlap of certain parts of the surface between the propagated Laplacian and the next frame. Such an example can be observed in Figure 6-f, where the left hand had the fist properly propagated (due to the protrusion region matching), but not the forearm (due to the lack of features). This caused the signed distance function based evolution to collapse a sub-part of the forearm and regrow it from the upper-arm and the fist. These rare cases can be addressed by interpolating the trajectories from the neighboring vertices which are tracked correctly.

If we are not satisfied with the propagated Laplacian, we can also try to increase the number of the matches by exploring the neighborhoods of the sparse matches detected by our method. These increased matches are then diffused in a similar fashion using the mesh Laplacian. At a minor additional computational cost, this process produces a better initialization for the mesh deformation step.

Another remark is that instead of surface morphing, one could also consider other functions. One such choice is multi-view stereo photo-consistency. We have experimented with such a distance function [8], observing that the optimizer could not easily handle situations where the source mesh is relatively far from the destination mesh. This is in part due to its coarse to fine nature. Another benefit of the mesh morphing approach is that, assuming there is some overlap between the source $\hat{\mathcal{S}}^t$ and the destination $\mathcal{M}^{t+1}$ meshes, it is guaranteed that the approach will converge, with potential topological changes. In addition, every vertex will reach the destination mesh. Once reached, it will neither move nor oscillate.

7 Results

7.1 Qualitative Evaluation



(a) pop2lock : frames 20-40 (b) pop2lock : frames 78-87 (c) flashkick : frames 50-60

(d) dance-2 : frames 567-582 (e) dance-2 : frames 620-635 (f) dance-2 : frames 607-622

Fig. 6. The tracked trajectories are presented in a color coded scheme where cooler colors represent earlier frames. On top the *pop2lock* and *flashkick* sequences (Univ. of Surrey) and bottom the *dance-2* sequence (INRIA).

For our evaluation we have been using sequences from two sources: the *dance-1* (used to exemplify the method) and the *dance-2* sequences are available publicly on our website [1]. The *pop2lock* and *flashkick* sequences were made available to us by the Surface Motion Capture project at the University of Surrey [29].

The *pop2lock* sequence provides us with full 3-D reconstruction results, together with the camera calibration, input images and silhouettes, using 8 cameras (1920x1080). The *dance-2* sequence is captured using 8 cameras (780x582). For this last sequence we decided to test the limits of the algorithm. We have used rougher 3-D surfaces approximation obtained via a fast visual hull reconstruction (exclusively based on silhouettes). Despite their coarse nature and topological changes, we still obtain consistent point trajectories. We ensured proper mesh sampling via edge collapses and edge swaps, such that each edge is around 3 pixels when projected onto the image (*pop2lock* meshes - 12,000 vertices; *dance-2* - 3,000 vertices). Coarser meshes were used for computing geodesics (1,500 vertices).

Our results are presented in Figure 6, with a close-up of a topological change illustrated in Figure 7. The tracks are color-coded, where cooler colors represent earlier frames. Additional convincing results are provided as a video [2], the natural way of displaying temporal information.

[1] https://charibdis.inrialpes.fr/html/sequences.php
[2] https://perception.inrialpes.fr/Publications/2008/VZBH08/ECCV08.mp4

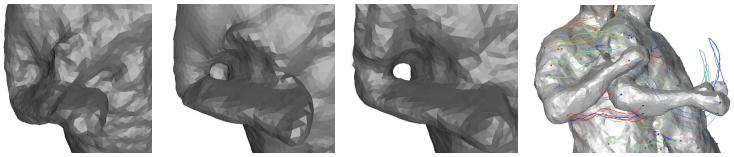
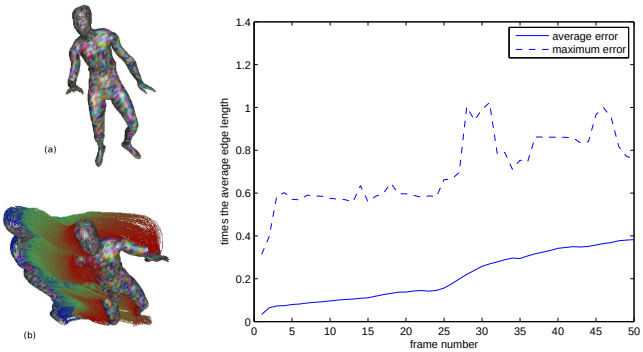


Fig. 7. Mesh deformation over topological change (from left to right): initialization; intermediate step; final step; Overall algorithm behavior (sparse set of matches shown only for ease of visualization purposes).

We were able to successfully track without problems *long* sequences of over 100 frames with *large* inter-frame shifts. The running times are satisfactory, depending a lot on the mesh density and the number of images used within each frame. As an example, in the *dance-2* sequence, an inter-frame surface tracking is produced in about 30 seconds, whereas for the *pop2lock* dataset, it takes about 2.5 minutes.

7.2 Numerical Evaluation



(c)

Fig. 8. Numerical Evaluation : (a) Example mesh with texture (b) Computed trajectories (c) Error over the sequence

Lack of proper ground truth makes quantitative assessment of 3D tracking algorithms difficult. A manual labeling could be inconsistent because the accuracy of the tracks needs to be measured with high precision. Due to the absence of real world test data, we evaluated the trajectories of our algorithm against known deformations of a 3D graphical model. We used an artificially textured female humanoid model (figure 8-a), and the multi-view video is captured using a 16 camera setup. An example trajectory computed by our algorithm is visualized in figure 8-b.

We evaluated the error in point trajectories with respect to the average edge length, which defines the resolution for temporal correspondences. Figure 8-c shows such errors for 600 points randomly distributed over the mesh and as obtained with independent estimations of the surface evolutions between frames. We observe that the error is less than half the average edge length after 50 frames. In the same duration, the average true deformation encountered by each point is about 10 times the average edge length. Thus we stay within reasonable limits of accuracy in producing our tracks.

8 Conclusion

In conclusion, we have presented a robust algorithm for temporal mesh tracking that incorporates the following key ingredients: it uses both geometric and photometric information in a coarse to fine fashion in order to efficiently solve for a sparse set of matches; it uses Laplacian propagation to obtain a dense match set; it ensures proper evolution using a mesh-morphing approach that is capable of dealing with topological changes. Thus, we are able to perform surface tracking with *large displacements* of surfaces with *topological changes* over *long sequences* in the context of multiple camera environments. In addition, our algorithm performs gracefully even when provided with inexact surfaces. For future work, we are considering real time motion capture systems, improvements of surface recovery, and reducing the drift in trajectories by time integration.

References

1. Gavrila, D., Davis, L.: 3-D model-based tracking of humans in action: a multi-view approach . In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, (USA). (1996) 1
2. Kakadiaris, I., Metaxas, D.: Model-based estimation of 3d human motion. IEEE Transactions on PAMI **22** (2000) 1453–1459 1
3. Carranza, J., Theobalt, C., Magnor, M., Seidel, H.P.: Free-viewpoint video of human actors. Proc. ACM Siggraph'03, San Diego, USA (2003) 569–577 1
4. DeCarlo, D., Metaxas, D.: Optical flow constraints on deformable models with applications to face tracking. International Journal of Computer Vision **38(2)** (2000) 99–127 1
5. Salzmann, M., J.Pilet, S.Ilic, P.Fua: Surface deformation models for non-rigid 3–d shape recovery. IEEE Transactions on PAMI **29** (2007) 1481–1487 1
6. Vedula, S., Rander, P., Collins, R., Kanade, T.: Three-Dimensional Scene Flow. IEEE Transactions on PAMI **27(3)** (2005) 474–480 1, 3
7. Neumann, J., Aloimonos, Y.: Spatio-Temporal Stereo Using Multi-Resolution Subdivision Surfaces. International Journal of Computer Vision **47** (2002) 181–193 1, 3
8. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. International Journal of Computer Vision **72(2)** (2007) 179–193 1, 3, 4, 10
9. de Aguiar, E., Theobalt, C., Stoll, C., Seidel, H.: Marker-less Deformable Mesh Tracking for Human Shape and Motion Capture. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, (USA). (2007) 1, 3, 5, 9

10. Anguelov, D., Srinivasan, P., Pang, H.C., Koller, D., Thrun, S., , Davis, J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In: Proceedings of Conference on Neural Information Processing Systems, Cambridge (USA). (2004) 1, 3
11. Bronstein, A., Bronstein, M., Kimmel, R.: Calculus of non-rigid surfaces for geometry and texture manipulation. IEEE Transaction on Visualization and Computer Graphics **13(5)** (2007) 902–913 1, 3
12. Starck, J., Hilton, A.: Correspondence labelling for wide-time free-form surface matching. In: Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, (Brazil). (2007) 1, 3
13. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer (2003) 2
14. Montagnat, J., Delingette, H., Scapel, N., Ayache, N.: Representation, shape, topology and evolution of deformable surfaces. application to 3d medical image segmentation. Technical Report 3954, INRIA (2000) 3
15. Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., Gross, M.: Multi-scale capture of facial geometry and motion. In: ACM Computer Graphics (Proceedings SIGGRAPH). (2007) 3
16. Carceroni, R., Kutulakos, K.: Multi-View Scene Capture by Surfel Sampling: From Video Streams to Non-Rigid 3D Motion, Shape and Reflectance. International Journal of Computer Vision **49(2-3)** (2002) 175–214 3
17. Hernandez, C., Schmitt, F.: Silhouette and stereo fusion for 3D object modeling. Computer Vision and Image Understanding **96** (2004) 367–392 3, 4
18. Furukawa, Y., Ponce, J.: Carved Visual Hulls for Image-Based Modeling. In: Proceedings of the 9th European Conference on Computer Vision, Graz, (Austria). (2006) 3, 4
19. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE Transactions on PAMI **14(2)** (1992) 239–256 3
20. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding **89(2-3)** (2003) 114 – 141 3
21. Zhang, D., Hebert, M.: Harmonic Maps and Their Applications in Surface Matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, (USA). (1999) 3
22. G, G.Z., R.Kimmel, Kiryati, N.: Texture mapping using surface flattening via multidimensional scaling. IEEE Transactions on Visualization and Computer Graphics **8(2)** (2002) 198–207 3
23. Starck, J., Hilton, A.: Spherical Matching for Temporal Correspondence of Non-Rigid Surfaces. In: Proceedings of the 10th International Conference on Computer Vision, Beijing, (China). (2005) 3
24. Sorkine, O.: Laplacian mesh processing. In: Eurographics Conference. (2005) 4, 8, 9
25. Zaharescu, A., Boyer, E., Horaud, R.: Transformesh: a topology-adaptive mesh-based approach to surface evolution. In: Proceedings of the 8th Asian Conference on Computer Vision, Tokyo (Japan). (2007) 4, 10
26. Bay, H., Tuytelaars, T., van Gool, L.: Surf : Speeded up robust features. In: Proceedings of the 9th European Conference on Computer Vision, Graz, (Austria). (2006) 5
27. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.: Topology matching for fully automatic similarity estimation of 3d shapes. In: ACM Computer Graphics (Proceedings SIGGRAPH). (2001) 5
28. Breen, D.E., Whitaker, R.T.: A level-set approach for the metamorphosis of solid models. IEEE Transaction on Visualization and Computer Graphics **7** (2001) 173–192 10
29. Starck, J., Hilton, A.: Surface capture for performance based animation. IEEE Computer Graphics and Applications **27(3)** (2007) 21–31 11