Dense Correspondence Finding for Parametrization-free Animation Reconstruction from Video

Naveed Ahmed, MPI Informatik
Christian Theobalt, Stanford University
Christian Rössl, Magdeburg University
Sebastian Thrun, Stanford University
Hans-Peter Seidel, MPI Informatik

Abstract

We present a dense 3D correspondence finding method that enables spatio-temporally coherent reconstruction of surface animations from multi-view video data. Given as input a sequence of shape-from-silhouette volumes of a moving subject that were reconstructed for each time frame individually, our method establishes dense surface correspondences between subsequent shapes independently of surface discretization. This is achieved in two steps: first, we obtain sparse correspondences from robust optical features between adjacent frames. Second, we generate dense correspondences which serve as map between respective surfaces. By applying this procedure subsequently to all pairs of time steps we can trivially align one shape with all others. Thus, the original input can be reconstructed as a sequence of meshes with constant connectivity and small tangential distortion. We exemplify the performance and accuracy of our method using several synthetic and captured real-world sequences.

1. Introduction

In recent years, ever more efficient computers and increasingly accurate imaging devices have rendered it feasible to capture computer animations from subjects performing in the real-world rather than by hand-crafting them with the traditional toolbox of the animator. To this end, a variety of methods have been developed that reconstruct both time-varying shape and appearance of arbitrary real-world performers from multi-viewpoint video, Sect. 2.

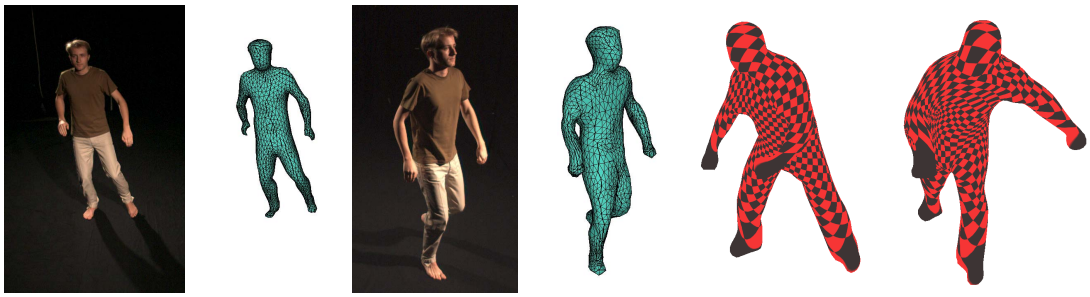
Most of these methods provide convincing shape and appearance for each time step of an input animation individually. However, they fall short of reconstructing spatio-temporally coherent scene geometry for arbitrary subjects since the challenging 3D correspondence problem is not addressed. Spatio-temporal coherence is an important and highly-desirable property in captured animations, as it greatly facilitates or even is inevitable for many tasks such as editing, compression or spatio-temporal postprocessing.

We therefore propose a new spatio-temporal dense 3D correspondence finding method that enables us to capture coherent dynamic scene geometry using standard shape-from-silhouette methods, Sect. 3. Our algorithm is tailored to the characteristics of video-based reconstruction methods which often capture high spatial detail in the input video frames, but provide relatively sparsely sampled 3D geometry with a much lower level of shape detail and with a considerable level of noise.

In a first step, shape-from-silhouette surfaces are reconstructed for each time step of video yielding a sequence of shapes made of triangle meshes with varying connectivity. Thereafter, sparse 3D correspondences between subsequent pairs of surfaces are computed by matching 3D positions of optical features that can be accurately extracted from high-resolution input video frames, Sect. 3.1. These sparse correspondences represent control points for anchoring appropriate bivariate scalar functions on each reconstructed surface mesh, Sect. 3.2. The choice of these functions enables us to establish dense correspondence essentially by matching function values. The dense correspondences can be used to straightforwardly align one mesh to all other reconstructions by performing a sequence of pairwise registrations, Sect. 3.3. The output of our approach is a spatio-temporally coherent animation, i.e. a sequence of meshes with constant graph structure and low tangential distortion.

2. Related Work

Technological progress in recent years has made it feasible to reconstruct shape and appearance of dynamic scenes using video [16] or video plus active sensing [28]. Multi-view video methods based on the shape-from-silhouette [17] or stereo principle [30] bear the intriguing advantage that they enable reconstruction of arbitrary moving subjects. Unfortunately, None of these methods is designed to reconstruct scene geometry with coherent connectivity over time since the 3D correspondence problem is not addressed. Model-based approaches employ shape priors [7, 6] which limits them to certain types of scenes. The algorithm proposed in this paper enables coherent dy-



(a) (b) (c) (d) (e) (f)

Figure 1. Input video frames (a), (c) and corresponding spatio-temporally coherent meshes rendered back into same camera view (b), (d). The checkerboard texture shows the consistently small tangential surface distortion in our reconstruction even between temporally far apart frames (e), (f). – See also accompanying video [1].

namic shape reconstruction while maintaining the flexibility of shape-from-silhouette methods.

In geometry processing, the 3D correspondence problem is addressed in parametrization and its application in (compatible) remeshing see, e.g., the surveys [12, 2] where the goal is to match the connectivity of one *single* shape model to the connectivity of another one. Generally, the required robust parametrization techniques are limited to fixed topology and are computationally involved, especially in the presence of additional constraints from given correspondences.

The key to spatio-temporally coherent reconstruction is a robust solution to the 3D correspondence problem. Conceptually similar to this problem, albeit in a reduced problem domain, is the shape matching problem [19]. One way to solve this problem is to localize and match salient geometric features between two shapes [10]. By combining feature matching with pose transformation, two shapes can be aligned [13]. Some probabilistic alignment methods register laser scans by finding the most probable embedding of one shape into the other [3]. Iterative closest point (ICP) procedures use a much simpler correspondence criterion that iteratively pairs locations closest to each other [11]. ICP methods may easily get stuck in local minima if no decent initial registration is provided. None of the aforementioned algorithms explicitly addresses the problem of multi-frame animation reconstruction.

Only few methods so far explicitly address the problem of reconstructing coherent animated surfaces from real-time scanner data, such as real-time structured light scanners [26, 24]. Unfortunately, in a video-based setting like ours, the applicability of these methods is either limited by high computational complexity, or by the requirement of high spatial and temporal sampling density which is typically not fulfilled.

Similar to our approach is the algorithm proposed by Shinya et al. [20] who deform a 3D model into sequences of visual hull meshes by minimizing a deformation energy. In contrast to our algorithm, accurate optical feature information is not exploited, and the ICP-like correspondence criterion is vulnerable to erroneous local convergence.

Matsuyama et al. [16] suggest a method to deform a mesh based on multi-view silhouettes and multi-view photo-consistencies. By optical means only, the required dense matches are difficult to find, and therefore the strongly constrained non-linear minimization takes several minutes computation time per frame. In contrast, our algorithm is computationally more efficient and creates dense correspondences despite only sparse optical matches.

Starck et al. [22] also aim at establishing coherence in sequences of shape-from-silhouette meshes. Their method establishes correspondences in a spherical parametrization domain which may fail in extreme poses and may introduce distortion-dependent matching inaccuracies close to singular points. In a recent follow-up, Starck et al. [23] apply a Markov random field to match isometry-invariant surface descriptors based on local parametrization. This enables establishing correspondence over wide time-frames, which is in fact a different problem. For both, [22, 23], numerical problems are more involved and computational costs are orders of magnitude higher [21] than for our method.

In contrast to the methods described above, our algorithm provides the following advantages and novelties

- As an object space method it does not suffer from parametrization-induced limitations.
- It establishes dense correspondence fields independently of the level and structure of surface discretization which makes surface alignment straightforward.
- It explicitly addresses the characteristics of shape-from-silhouette-based animation reconstruction. By combining both accurate image feature and function matching, we are able to robustly match even coarsely reconstructed surface geometry lacking coherent and dense surface details.
- In practice, robustness to topology changes.

3. Spatio-temporal Correspondence Finding

The input to our method is a sequence of calibrated synchronized video streams that were recorded from multiple viewpoints around the scene and that show a subject performing in the scene's foreground. Our test acquisition system features eight synchronized video cameras arranged in a circular setup and delivering 25fps at 1004x1004 pixel frame resolution.

Background subtraction yields a foreground silhouette for each of the N captured video frames. In a preprocessing step a polyhedral visual hull method [9] is applied to each time-step of video. In order to cure triangle degeneracies in the input data and to produce a more uniform surface discretization, the visual hull surfaces are resampled and the resulting point clouds are fed into a Poisson surface reconstruction approach [14] (we use their implementation). This way, a sequence of triangle meshes with varying vertex connectivity is produced that captures the shape of the subject at each time step.

In the following we describe a triangle mesh as $\mathcal{M} = (\mathcal{V}, \mathcal{T}, \mathbf{p})$, where $\mathcal{V}$ denotes vertices and $\mathcal{T}$ their triangulation or *connectivity*. Hence, $(i, j, k) \in \mathcal{T}$ denotes a triangle, and with each vertex $\ell \in \mathcal{V}$ we associate positions $\mathbf{p}_\ell \in \mathbb{R}^3$ defining the surface's embedding in 3D. We consider N time-frames and thus write a sequence of meshes as $\mathcal{M}(t) = (\mathcal{V}(t), \mathcal{T}(t), \mathbf{p}(t)), t = 0, \ldots, N-1$, where $\mathcal{M}(t)$ approximates the (ideal) surface $\mathcal{S}(t)$.

Our algorithm propagates the connectivity of mesh $\mathcal{M}(0)$ by iteratively matching it against reconstructed visual hull meshes. In the following, we write $\mathcal{M}_0(t)$ for meshes with connectivity $(\mathcal{V}_0, \mathcal{T}_0) := (\mathcal{V}(0), \mathcal{T}(0))$ of $\mathcal{M}(0)$, i.e., $\mathcal{M}_0(t) = (\mathcal{T}(0), \mathcal{V}(0), \mathbf{p}(t))$ and in particular $\mathcal{M}(0) = \mathcal{M}_0(0)$. Then given a subsequent pair of meshes $\mathcal{M}_0(t)$ and $\mathcal{M}(t+1)$, where $\mathcal{M}_0(t)$ is $\mathcal{M}(0)$ aligned with $\mathcal{M}(t)$ during a previous iteration, our algorithm proceeds as follows:

In a first step, initial coarse correspondences are obtained by matching robust optical features between image-frames and mapping them to 3D-positions on the surfaces, Sect. 3.1. We use SIFT [15] for this purpose, yielding a sparse covering of the surfaces with feature points. In contrast to deformation transfer methods [25, 29], we can't choose ideal features, i.e. our sparse features alone generally don't carry enough information for direct correspondence or deformation-based alignment, see also Sect. 5.

Therefore, we estimate dense correspondences in a second step, which constitutes the core of our approach: with each feature point we associate a scalar, monotonic function with certain interpolation properties. Requirements for such functions will be discussed in detail in Sect. 3.2. Dense correspondences are found by pairing surface locations with similar function values.

This way we can provide surface correspondences which are densely and faithfully distributed over the surface. We use these matching 3D surface points as constraints for deforming one mesh over time without resorting to involved deformation algorithms (see, e.g., [5]) that were necessary if correspondences were sparse. The result is an animation sequence with constant connectivity.

We remark that the approach is tailored to the particular animation setting: the acquisition and shape-from-silhouette reconstruction provides only fairly accurate and medium resolution geometry data, possibly contaminated with noise, but at the same time high-resolution texture information per image frame. The individual matching steps are detailed in the following subsections.

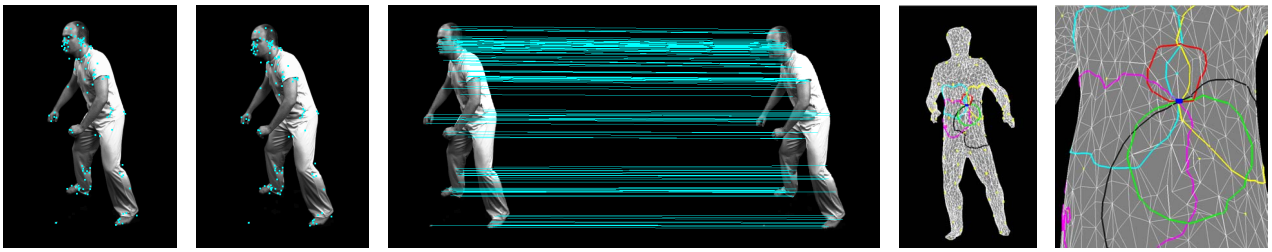
3.1. Coarse Correspondences

In order to establish coarse correspondences we find robust optical features between adjacent frames by localizing them in the input video frames and inferring their 3D positions by means of the available reconstructed model geometry. For localizing features we apply SIFT descriptors [15] as this technique has a number of advantageous properties for our video setting: identified features are largely invariant under rotation, scale and moderate change in viewpoint, and the rich descriptors also enable wide-baseline matching. In particular the latter property pays off in our setting as rapid scene motion may easily lead to large image disparities between subsequent frames. In such a scenario, alternative image matching approaches, such as KLT or general optical flow methods are more likely to fail [4]. Also, as opposed to geometric feature matching [10] we can maintain precision even if the reconstructions don't exhibit salient shape details.

We compute 2D SIFT feature locations for each input frame $I_c(t)$ at all time steps t and all camera views c in a preprocessing step. On a typical sequence we obtain between 300 and 500 features per time step (with multiple occurrences of the same feature across cameras discarded.

When aligning two subsequent meshes $\mathcal{M}_0(t)$ and $\mathcal{M}(t+1)$, we compute 3D feature positions at either time step by back-projection from images onto the 3D shapes. To preserve the highest possible feature localization accuracy independently of triangulation (from Marching Cubes after Poisson reconstruction), 3D positions of features are computed from linear interpolation rather than nearest vertex positions. To this end, we exploit the graphics hardware and assign to each feature an interpolated 3D position obtained via rasterizing the 3D shape's coordinates into the same camera view.

To facilitate later computation of dense correspondences, we intermediately enforce association of features with vertices by locally splitting each original triangle containing a feature into three triangles. This is achieved by inserting a new vertex at the interpolation point. By performing 3D



(a) (b) (c) (d) (e)

Figure 2. Detected SIFT features in two consecutive frames (a) and (b). Matched features are shown in (c). Obvious outliers, such as matches outside the silhouette, are filtered out during preprocessing. Intersecting iso-contours of harmonic functions centered on sparse correspondences (shown as colored lines) can be used to localize surface points. For clarity, (e) zooms in on a subregion of (d).

localization and subdivision for all camera views at a each time step t and $t+1$, we create a set of possibly subdivided versions of the original reconstruction meshes $\mathcal{M}'_0(t)$ and $\mathcal{M}'(t+1)$. Each of these meshes possesses an associated set of feature vertex indices $\mathcal{F}(t)$ and $\mathcal{F}(t+1)$. Note that these meshes only serve as temporary helper structures to gain accuracy. Local splits will be rolled back later, and are neither used in the final output of our method nor induce any other side effects, see Sect. 3.3. Therefore, and to keep notation simple, we will continue to refer to $\mathcal{M}_0$ and $\mathcal{M}$.

We find correspondences between SIFT feature vertices on either mesh by looking for pairs with similar descriptors. To this end, we compute the Euclidean distance $D_e(i,j)$ between the descriptors of all elements $i \in \mathcal{F}(t)$ and $j \in \mathcal{F}(t+1)$. A correspondence (i,j) is considered plausible and hence established if $D_e(i,j)$ is below a certain threshold. This way, possible outliers in all correspondence sets are filtered out by discarding matches with implausible 3D distances. Erroneous matches outside the silhouette area are trivially discarded. Fig. 2(a-c) illustrates SIFT features.

3.2. Finding Dense Correspondences

The basic idea for establishing dense correspondence is to infer additional values from the given sparse features and the surface, and to then carefully analyze and compare these values over time. For this purpose we define bivariate scalar functions h_i on the surfaces, each function is associated with a particular feature $f_i \in \mathcal{F}$, $i = 0, \ldots, m$. In an ideal setting we could think of these as distance or coordinate functions: given three (feature) points a, b, c in the plane, any point in the plane can be characterized by its distance to each of a, b, c or in terms of its barycentric coordinates w.r.t. the triangle (a, b, c). Our choice of functions h_i resembles barycentric coordinates as we require *interpolation* $h_i(\mathbf{u}_i) = 1$ and $h_i(\mathbf{u}_j) = 0$ for all $i \neq j$, and *monotonicity* of h_i with extrema at the interpolation points, where $\mathbf{u}_i \in \mathbb{R}^2$ denotes a surface point associated with f_i.

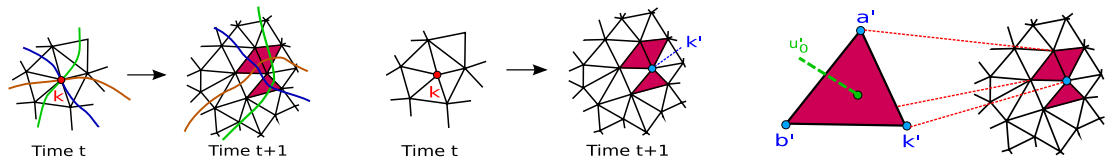
In order to be meaningful when evaluated for different t over the time-dependent surface $\mathcal{S}(t)$, we additionally require that h_i is taken from a class of functions which change their values only slightly under moderate surface deformations. For this reason we chose harmonic functions which satisfy

$$\Delta_{\mathcal{S}(t)}\, h_i = 0 \ , \tag{1}$$

where $\Delta_{\mathcal{S}(t)}$ denotes the Laplace-Beltrami operator. This is justified by the isometry-invariance of the operator, i.e., for isometric deformations of $\mathcal{S}$ into $\mathcal{S}'$ we have $\Delta_{\mathcal{S}} = \Delta_{\mathcal{S}'}$. We assume moderate deformations of $\mathcal{S}(t)$ to be largely isometric. This property has previously been exploited to compute signatures for shape matching and retrieval, see, e.g., [8, 18].

So far we assumed continuous functions. In practice, h_i are piecewise linear functions w.r.t. $\mathcal{M}(t)$, and an appropriate discretization of the differential operator $\Delta_{\mathcal{S}(t)}$ is required. In particular, we require independence of the triangulation, i.e. for different meshes approximating the same shape, the discrete solutions of (1) should yield the same or very similar results. We use the well-established cotangent discretization which provides this linear-precision property and is symmetric (see [27] for a comparison of alternative discretizations).

With functions h_i computed we proceed in several steps to find dense correspondence. Given a surface point $\mathbf{u}_0 \in \mathcal{S}(t)$ that corresponds to a vertex k of $\mathcal{M}_0(t)$, the goal is to find a matching point $\mathbf{u}'_0 \in \mathcal{S}(t+1)$ using h_i defined on the mesh $\mathcal{M}_0(t)$ and h'_i defined on $\mathcal{M}(t+1)$. Evaluation of the harmonic functions yields "coordinates" $\mathbf{h}(\mathbf{u}) := [h_0(\mathbf{u}), \ldots, h_m(\mathbf{u})]$ and $\mathbf{h}'(\mathbf{u}) := [h'_0(\mathbf{u}), \ldots, h'_m(\mathbf{u})]$ for both surfaces. As contributions of $\mathbf{h}$ are localized we restrict ourselves to the K coordinate values of largest magnitude at $\mathbf{u}_0$, i.e., we consider $\mathbf{h}_{\mathbb{K}}(\mathbf{u}_0) := [h_{i_1}, \ldots, h_{i_K}]$, $i_1, \ldots, i_K \in \mathbb{K}$, where $h_\ell(\mathbf{u}_0) \geq h_{\overline{\ell}}(\mathbf{u}_0)$ for all $\ell \in \mathbb{K}, \overline{\ell} \notin \mathbb{K}$. In our implementation we use $K = 10$. We can visualize the local influence of the h_i geometrically by the analog of a planar Voronoi diagram thinking of $1 - h_i$ as distance function. Then for each element in a "Voronoi cell", we expect significant or meaningful contribution only from functions associated with the cell and its immediate neighbor cells. We therefore chose K conservatively, as on average one will find 6 immediate neighbors. In an ideal setting,



(a) (b) (c)

Figure 3. (a) Vertex k (corresponding to $\mathbf{u}_0$) and the iso-contours intersecting at it. For better visibility only $K = 3$ contours are shown. At time $t + 1$, the same iso-contours don't intersect in a single point. Each candidate triangle (shown in red) is intersected by two of the iso-contours. (b) A vertex k' from the candidate triangle set on $\mathcal{M}(t + 1)$ that is closest to k according to D_h criterion is selected. (c) Finding the surface point $\mathbf{u}'_0$ within the best-matching triangle (a', b', k') (according to D_h) that is adjacent to k'.

$\mathbf{h}(\mathbf{u}) = \mathbf{h}'(\mathbf{u})$, and retrieving $\mathbf{u}'$ can be imagined as intersecting iso-contours $h'_i(\cdot) = h_i(\mathbf{u}_0)$, $i \in \mathbb{K}$. Fig. 2(d),(e) illustrates this concept by visualizing several iso-contours on the surface of a visual hull mesh intersecting in a single vertex. In the presence of moderate deformations and given discrete meshes, the equality generally does not hold. Therefore, instead of exact intersections, we are interested in a set of *triangles* $\mathcal{E} \subset \mathcal{T}(t+1)$, which are intersected by at least one of the iso-contours passing through $\mathbf{u}_0$. These are triangles in which $\mathbf{u}'_0$ potentially resides. To put this idea into practice, we add to $\mathcal{E}$ all those triangles that are intersected by the highest number of contours with iso-value $h_i(\mathbf{u}_0)$. This yields a (potentially) 1-to-many match from $\mathbf{u}_0$ to a set of candidate triangles, see Fig. 3(a). To handle possible localization inaccuracies, in practice we build $\mathcal{E}$ conservatively and also include all candidate triangles for the vertices in a 1-ring around $\mathbf{u}_0$ which are identified by the same procedure.

To determine the final position of $\mathbf{u}'_0$ on $\mathcal{M}(t + 1)$, we first identify the vertex $k' \in \mathcal{V}_{t+1}$ that is closest to $\mathbf{u}'_0$. We extract this vertex k' from the set $\mathcal{E}$ by computing a distance measure between $\mathbf{h}_{\mathbb{K}}(\mathbf{u}_0)$ and $\mathbf{h}'_{\mathbb{K}}(\mathbf{u}'_\ell)$ for all vertices ℓ out of $\mathcal{E}$, see Fig. 3(b) for illustration on a simplified setting. (Note that the set $\mathbb{K}$ is determined w.r.t. $\mathbf{h}$ on $\mathcal{M}_0$.)

Through experiments we found the following measure to work very satisfactorily. Let $\mathbf{d}_{\mathbb{K}} := \mathbf{h}_{\mathbb{K}}(\mathbf{u}_0) - \mathbf{h}'_{\mathbb{K}}(\mathbf{u}'_\ell)$. We define the distance $D_h(\mathbf{u}_0, \mathbf{u}'_\ell)$ as

$$D_h(\mathbf{u}_0, \mathbf{u}'_\ell) \;=\; \mathbf{d}_{\mathbb{K}} \; (\mathbf{I} - \mathrm{diag}(\mathbf{h}'_{\mathbb{K}}(\mathbf{u}'_\ell))^3 \; \mathbf{d}_{\mathbb{K}}^\top \, .$$

Let $\mathcal{E}_{\mathcal{V}}$ contain all vertices shared by triangles in $\mathcal{E}$. We select that vertex $k' \in \mathcal{E}_{\mathcal{V}}$ with minimal distance, i.e. $D_h(\mathbf{u}_0, \mathbf{u}'_{k'}) \leq D_h(\mathbf{u}_0, \mathbf{u}'_\ell)$ for all $\ell \neq k', \ell \in \mathcal{E}_{\mathcal{V}}$.

The final step in finding $\mathbf{u}'_0$ is to localize its position at sub-discretization accuracy since, in general, $\mathbf{u}'_0$ is an arbitrary surface point and won't coincide with a vertex location. To achieve this purpose, we first identify the triangle (a', b', k') in the 1-ring of k' for which the average of $D_h(\mathbf{u}_0, \mathbf{w})$ (with $\mathbf{w} \in \{\mathbf{u}_{a'}, \mathbf{u}_{b'}, \mathbf{u}_{k'}\}$) is minimal. The best-matching surface point is expressed linearly as $\mathbf{u}'_0 = \lambda_{a'} \mathbf{u}_{a'} + \lambda_{b'} \mathbf{u}_{b'} + \lambda_{k'} \mathbf{u}_{k'}$. We determine $\mathbf{u}'_0$ within (a', b', k) as

$$\underset{\lambda_{a'}, \lambda_{b'}, \lambda_{k'}}{\arg\min} \; ||\mathbf{d}_{\mathbb{J}}||^2 \, ,$$

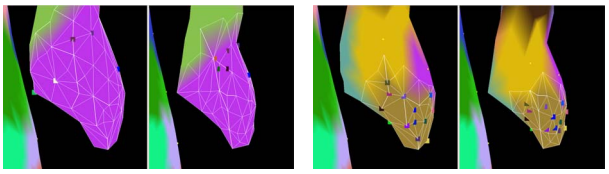
where $\mathbf{d}_{\mathbb{J}} := \mathbf{h}_{\mathbb{J}}(\mathbf{u}_0) - \mathbf{h}'_{\mathbb{J}}(\mathbf{u}'_0)$ and $\mathbb{J} \subset \mathbb{K}$ contains the indices of the three largest coordinate values at $\mathbf{u}_0$. Intuitively, we thereby place $\mathbf{u}'_0$ as close as possible to either of the three highest-value iso-contours within the area of (a', b', k'), ideally at their intersection point. Fig. 3(c) illustrates this last step.

3.2.1 Remarks on practical implementation

Computation of coordinate functions. Numerically, h_i can be computed for every $\mathcal{M}(t)$ very efficiently by factoring a sparse matrix and then applying $m + 1$ back-substitutions. As a result we obtain $m + 1$ linear functions h_i , i.e., for every vertex $j \in \mathcal{V}$ we have $h_i(\mathbf{u}_j)$. In practice, we compress this data efficiently by storing only the K largest values together with associated feature indices $\mathbb{I}_j = \{i_1, \ldots, i_K\} \subset \mathcal{F}$. Hence, for every vertex j we store $h_\ell(\mathbf{u}_j)$, $\ell \in \mathbb{I}_j$, where $h_\ell(\mathbf{u}_j) \geq h_{\overline{\ell}}(\mathbf{u}_j)$, $\overline{\ell} \notin \mathbb{I}_j$. Consequently, we implicitly assume $h_{\overline{\ell}}(\mathbf{u}_j) = 0$, which is reasonable and induces only small error as the values of h_i fall off quickly and significant contribution is localized. This way, we never require more storage than for $(K+1) \times \#\mathcal{V}$ values and indices for the cost of $\#\mathcal{V}$ K-element sorts after each solution of the Laplace equation.

Intersection with iso-contours. The intersections of triangles with an iso-contour $h_i(\mathbf{u}) = c$ can be implemented by a local search without additional data structures: Starting from the vertex associated with the feature f_i, i.e. where $h_i(\mathbf{u}_i) = 1$, we apply a gradient descent (h_i is monotone) on an arbitrary triangle attached to this vertex. We keep descending neighboring triangles until we hit a triangle that is intersected by the iso-contour. We then iteratively traverse all neighboring triangles which are also intersected.

Prefiltering of SIFT features and adaptive refinement. Coarse correspondences identified in Sect. 3.1 may be distributed unevenly on the surface and can therefore be redundant if concentrated in certain areas. We can exploit this redundancy and reduce computation time by prefiltering keeping only a well-distributed subset. To identify the active feature subset, we partition the surface into patches with similar geodesic radius or geometric complexity. For



(a) (b)

Figure 4. Feature prefiltering and refinement. (a) zoom-in onto hand region of the model at two subsequent time steps. Colored areas represent surface regions. Due to sparse distribution of coarse features, the correspondences (colored dots) are not correct. (b) Adaptively increasing the number of coarse features leads to accurate correspondences.

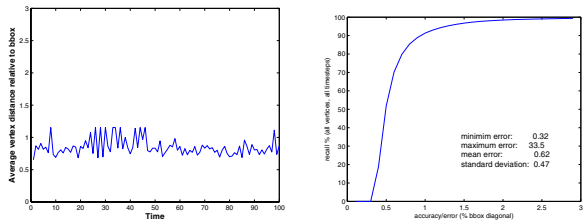
each resulting surface cell, we maintain only one coarse feature (colored regions in Fig. 4(a)). In local sub-regions this reduction of coarse correspondences may lead to too few adjacent "cells" to yield meaningful coordinates. There we raise the number of coarse correspondences, thereby adaptively increase the patch density and then proceed iteratively as described above. Fig. 4(b) shows that – on this particular data set – the latter greatly improves matching robustness in the hand region of the reconstructed human.

3.3. Alignment by Deformation

One intriguing advantage of our approach is that in the ideal case the dense correspondence field specifies the complete alignment of $\mathcal{M}_0(t)$ and $\mathcal{M}(t+1)$. To register the two meshes, we can therefore trivially move vertex locations without having to resort to involved deformation schemes. In practice, we find it advantageous to apply a fast and simple Laplacian deformation scheme rather than to perform vertex displacements only. This setting allows for trivial enforcement of surface smoothness during alignment hence smoothing out noise and mismatches. We refer to the recent survey [5] and the references therein for information on the method and its many variants. Laplacian deformation helps us to cure local reconstruction inaccuracies which may occur in surface regions for which feature localization was non-trivial, e.g. due to texture uniformity. Also, we take care that no loss of volume is introduced by the latter deformation approach: in rare cases where this becomes necessary, we force vertices of $\mathcal{M}_0(t)$ back onto $\mathcal{M}(t+1)$ along the shortest distance. This way we effectively deform $\mathcal{M}_0(t)$ to time-step $t+1$, and as we iterate the whole matching process over time, we track a single consistent mesh over the whole sequence, see Fig. 1 and Fig. 7

4. Results

To demonstrate the performance of our reconstruction approach, we recorded two real-world motion sequences in our multi-camera system. The first sequence comprising of 105 frames shows a walking subject, Fig. 1(a)-(d), and the second sequence comprising of 100 frames shows a human



(a) (b)

Figure 5. (a) Average vertex distance (in $\mathbb{R}^3$) over time. (b) Recall accuracy (geodesic) for all vertices in complete sequence. Errors given w.r.t. ground truth sequence in % of bounding box size (1% error ~ 1.8 cm)

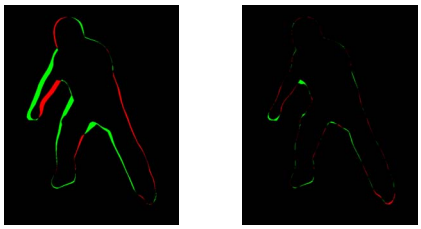
performing a simple capoeira move, Fig. 7. As shown in these images as well as the accompanying video [1], our method enables faithful reconstruction of spatio-temporally coherent animations from this footage. A side-by-side comparison of the original input sequence and the reconstructed mesh sequence shows that our method delivers coherent scene geometry with low tangential distortion. When texturing our result with a fixed checkerboard, coherence and low distortion properties become very obvious, see Fig. 1(e),(f) and the **accompanying video [1]**. We chose this visualization as texturing with the input video images would hide any geometric distortions.

Our algorithm is computationally more efficient than most deformation-based registration methods (see Sect. 2). Even if very detailed meshes comprising of roughly 10,000 vertices are reconstructed (Fig. 7(a)-(d)) and almost 600 coarse features are used, correspondences between pairs of frames can be computed in approximately 2 minutes on a Pentium IV 3.0 GHz. Prefiltering and adaptive refinement down to 120 coarse matches reduces alignment time to 1 minute per frame. In the more likely and practical case that mesh complexity is around 400 vertices, two frames can be aligned in as fast as 2 seconds even without prefiltering.

Even if surface triangulations are very coarse, our method produces high-quality coherent mesh animations and the advantages of the coherent mesh representation become even more evident. In the non-coherent version large triangulation differences between adjacent frames, Fig. 7(g),(h), lead to strong temporal noise which is practically eliminated in the coherent reconstructions, Fig. 7(e),(f).

5. Evaluation and Discussion

In order to measure the accuracy of our algorithm we created a synthetic ground truth video sequence by texturing a virtual human character model (skeleton+surface mesh) with a constant noise texture, animating the model with captured motion data, and rendering it back into 16 virtual camera views. By this means, we obtain for each



(a) (b)

Figure 6. Overlap of silhouettes of input and reprojected reconstructions in one camera view (red: non-overlapping pixels of input silhouette; green: non-overlapping pixels of reconstruction). (a) Coarse correspondences alone don't lead to a satisfactory alignment. (b) Dense correspondences, however, lead to an almost perfect alignment.

time step a ground truth 3D model with constant triangulation, as well as respective image data. To compare our results against ground truth, we reconstruct visual hull meshes for all frames of the synthetic input and align the ground truth 3D model of the first frame with all subsequent ones. Fig. 5(a) shows that the average vertex distance between the ground truth and the coherent reconstruction remains at a very low level of 1% of the bounding box dimension over time. The plot also shows no significant error drift which underlines the robustness of our algorithm. Fig. 5(b) shows recall accuracy: for more than 90% of the vertices (all time-steps) we are within 1% bounding box diagonal ($<$ 2cm) error radius.

By comparing the overlap between the coherent animations and the input silhouette images, we can assess the reconstruction quality of real sequences. On average, around 2.4% of the input silhouette pixels do not overlap with the reprojection which corresponds to an almost perfect match between input and our result, see Fig. 6(b). This comparison also clearly shows that dense correspondences are indeed needed to achieve this quality level as a deformation based on coarse features alone leads to a high residual alignment error, Fig. 6(a).

Our visual and quantitative results confirm effectiveness and efficiency of our method. In the following we discuss some properties and limitations inherent to the approach.

As we reconstruct shape from silhouette in every frame, the quality of results depends on the quality of the input video data and may suffer from artifacts attributed to the visual hull method itself. Some of the apparent phantom volumes in the results are solely due to the inability of shape-from-silhouette method to reconstruct concavities, and they are not introduced by our correspondence method. The focus of this paper is not improving per-time step shape-reconstruction itself, and our method could be used in just the same way with more advanced reconstruction methods that also enforce photo-consistency, such as space carving.

Comparing to related work by Starck et al. [22], our approach is more flexible (handles surfaces of arbitrary genus) and more efficient [21] as it does not rely on spherical parametrization, which is a non-trivial problem in its own. For their recent follow-up paper [23], we first remark that their goal is different in that wide time-frames are taken into account to solve a global problem. Hence, it is natural that our local approach is much more efficient. At the same time is accurate (they report typical errors of 5–10cm in their setting) and provides a map for *any* surface point.

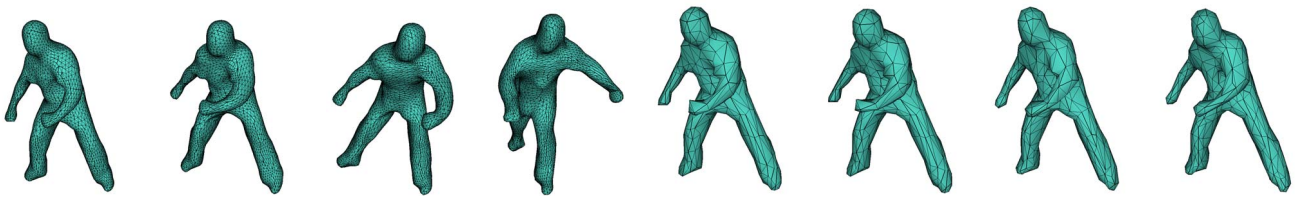
Also, some video sequences show a fair amount of motion blur, and hence some reconstruction errors appear which could be easily overcome with faster cameras. Despite these unfaithful reconstructions our tests show the robustness of our method.

Our approach does not require surface parametrization. However, it shares one limitation with most practical parametrization methods, namely the absence of guarantees to obtain a valid one-to-one mapping: this means local fold-overs may occur when triangles are mapped between surfaces [12]. In practice, the alignment by means of Laplacian deformation smoothes out such local mismatches. This fact and experiments back the assumption of nearly isometric deformations.

From a theoretical point of view our method is not proven to handle changes of the surface topology over time: "coordinate" functions might be locally unrelated in this situation, hence there is no guarantee that results are meaningful in the affected surface regions. Note that similar arguments are true for *any* method relying on local isometry which is not given under topology changes. In practice however, our method performs robustly towards typically observed topology changes (such as arms and legs merging in the visual hulls) similarly to [23]. To illustrate this robust handling, the video contains two synthetically generated example sequences (similar to the sequence used for accuracy measurement) in which arms and legs merge with the rest of the body. Generally, our goal is spatio-temporally coherent reconstruction, hence, topology changes should be avoided or corrected during the initial reconstruction step.

We gave intuitive motivation for selecting suitable "coordinate" functions and applying appropriate matching of surface points. We should remark that several aspects of our approach are based on heuristics which are justified only empirically, in particular the choice of distance measure D_h. An alternative approach might be based on learning techniques which compute perfectly parametrized distance functions for training sets.

Despite these limitations we have presented a robust and efficient dense correspondence finding method that enables spatio-temporally coherent animation reconstruction from multi-view video footage.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7. (a)-(d) Sample frames from a spatio-temporally coherent reconstruction of a capoeira move. Note that the actor's shape is faithfully reconstructed and triangle distortions are low. Remaining geometry artifacts are solely due to limitations of shape-from-silhouette methods. – The advantage of our reconstruction becomes very apparent in case of coarse triangulations (~ 750 triangles). (e), (f) show subsequent frames from our reconstruction, and (g),(h) the same frames from the non-coherent input. The triangulation in the former models remains very consistent while in the latter case the triangulation dramatically changes even from one time step to the next.

6. Conclusions

We presented a method to establish dense surface correspondences between originally unrelated shape-from-silhouette volumes that have been reconstructed from multi-view video. Our approach relies on sparse robust optical features from which dense correspondence is inferred in a discretization-independent way and without the use of parametrization techniques. Dense correspondences serve as maps between surfaces to align a mesh with constant connectivity to all per-time-step reconstructions. Our experiments confirm efficiency and robustness of our approach, even in the presence of topology changes. As results we reconstruct animations from video as a deforming mesh with constant structure and low tangential distortion. This kind of input is required by subsequent higher-level processing tasks, such as analysis, compression, reconstruction improvement, etc., which we would like to further explore and adapt in future work.

References

[1] http://www.mpi-inf.mpg.de/∼nahmed/CVPR08a.wmv .

[2] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring*. Spinger, 2007.

[3] D. Anguelov, D. Koller, P. Srinivasan, S. Thrun, H.-C. Pang, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Proc. NIPS*, 2004.

[4] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt. Performance of optical flow techniques. In *CVPR*, pages 236–242, 1992.

[5] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE TVCG*, 2007. To appear.

[6] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette for articulated objects and its use for human body kinematics estimation and motion capture. In *Proc. CVPR*, 2003.

[7] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *Proc. CVPR*, pages 1–8. IEEE, 2007.

[8] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Trans. PAMI*, 25(10):1285–1295, 2003.

[9] J.-S. Franco and E. Boyer. Exact polyhedral visual hulls. In *Proc. of BMVC*, pages 329–338, 2003.

[10] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM TOG*, 25(1):130–150, 2006.

[11] D. Hähnel, S. Thrun, and W. Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proc. of IJCAI*, 2003.

[12] K. Hormann, B. Levy, and A. Sheffer. Mesh parameterization: Theory and practice. In *SIGGRAPH Course Notes*, 2007.

[13] D. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *IVC*, 21(7):637–650, July 2003.

[14] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. SGP*, pages 61–70, 2006.

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE ICCV*, volume 2, page 1150ff, 1999.

[16] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation and high fidelity visualization for 3d video. *CVIU*, 96(3):393–434, 2004.

[17] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Proc. EGRW*, pages 116–126, 2001.

[18] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-beltrami spectra as 'shape-DNA' of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[19] S. Rusinkiewicz, B. Brown, and M. Kazhdan. 3d scan matching and registration. In *ICCV short courses*, 2005.

[20] M. Shinya. Unifying measured point sequences of deforming objects. In *Proc. of 3DPVT*, pages 904–911, 2004.

[21] J. Starck. personal communication.

[22] J. Starck and A. Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. *IEEE ICCV*, pages 1387–1394, 2005.

[23] J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *IEEE ICCV*, 2007.

[24] C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, and H.-P. Seidel. Template deformation for point cloud fitting. In *Proc. SGP*, pages 27–35, 2006.

[25] R. W. Sumner and J. Popovic. Deformation transfer for triangle meshes. *ACM TOG (Proc. SIGGRAPH)*, 23(3):399–405, 2004.

[26] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Proc. SGP*, pages 49–58, 2007.

[27] M. Wardetzky, S. Mathur, F. Klberer, and E. Grinspun. Discrete Laplace operators:no free lunch. In *Proc. SGP*, pages 33–37, 2007.

[28] M. Waschbuesch, S. Wuermlin, and M. Gross. 3D video billboard clouds. In *Proc. Eurographics*, 2007.

[29] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. *Computer Graphics Forum*, 24(3):601–609, 2005.

[30] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM TOG (SIGGRAPH)*, 23(3):600–608, 2004.