

# Silhouette Based Generic Model Adaptation for Marker-Less Motion Capturing<sup>\*</sup>

Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel

Max-Planck-Institut für Informatik,  
Stuhlsatzenhausweg 85,  
66123 Saarbrücken, Germany

**Abstract.** This work presents a marker-less motion capture system that incorporates an approach to smoothly adapt a generic model mesh to the individual shape of a tracked person. This is done relying on extracted silhouettes only. Thus, during the capture process the 3D model of a tracked person is learned.

Depending on a sparse number of 2D-3D correspondences, that are computed along normal directions from image sequences of different cameras, a Laplacian mesh editing tool generates the final adapted model. With the increasing number of frames an approach for temporal coherence reduces the effects of insufficient correspondence data to a minimum and guarantees smooth adaptation results. Further, we present experiments on non-optimal data that show the robustness of our algorithm.

## 1 Introduction

We address the problem of human shape and motion capture (MoCap) from multi-view video sequences. Surveys on these topics can be found in [15, 14]. Approaching marker-less methods, researchers working in the area of computer vision typically prefer simplified human body models [4, 13, 10, 11]. There are also methods in the field of Computer Graphics [12, 6, 24]. However, techniques for image processing or pose estimation are often oversimplified.

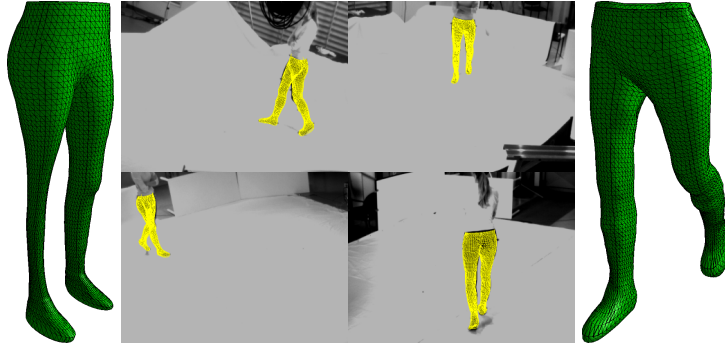
Cheung et al. [8] propose a shape-from-silhouettes approach that is applied to track human beings and incorporates surface point clouds with skeleton models. A work of Rosenhahn et al. [18] combines silhouette based pose estimation with more realistic human template models. These are represented by free-form surface patches and gain more accurate tracking results.

In order to allow for individual shapes during MoCap, Mündermann et al. [16] propose a MoCap system that combines the tracking algorithm with a database of articulated 3D models. The models are generated from a template mesh given by a deformable human model that is learned from a database of full body laser scans. During MoCap they select the best fitting 3D model from their database.

Bălan et al. [5] follow an approach of immediately incorporating a low-dimensional deformable human body model (SCAPE) into MoCap. Their idea

---

<sup>\*</sup> We gratefully acknowledge funding by the Max-Planck Center for Visual Computing and Communication.



**Fig. 1.** A multi-view image sequence: *Left:* Generic model, *Middle:* Tracked adapted model within image sequence, *Right:* Adapted model in tracked pose

is to learn pose and detailed shape by a stochastic optimization function that estimates the model parameters directly from 2D image data.

Another approach to jointly capture human motion and shape is presented by De Aguiar et al. [9]. They make use of a high-quality laser scan of the person to track and combine an image-based 3D correspondence estimation algorithm with a fast Laplacian mesh deformation scheme.

The proposals of [9, 5, 16] have in common, that there is the need for 3D laser scans. The latter ones even require special databases. Thus, the actual MoCap process comes along with extra costs. In contrast to these approaches, we present an algorithm that extends a marker-less MoCap system by an iterative adaptation algorithm. By the use of silhouette information, it smoothly adapts a generic template model to the true shape of the tracked person. Depending on a sparse number of 2D-3D correspondences, that are computed along normal directions from image sequences of 4 cameras, a Laplacian mesh editing tool generates the final adapted model. With the increasing number of frames an approach for temporal coherence reduces the effects of insufficient correspondence information to a minimum. As a result our algorithm increases tracking accuracy.

This work is built upon the basic tracking system and foundations in Section 2. This includes techniques for image segmentation with level sets, pose estimation of kinematic chains and shape registration based on ICP. Focus of this work is embedding the silhouette-based model adaptation algorithm. It is described in Section 3. Section 4 presents experiments as well as their results, and Section 5 concludes this work.

### 1.1 Contributions

This work contributes a method to compute accurate and smooth 3D models from a generic template. That is done during a silhouette based, marker-less MoCap process. In a temporal coherent approach we apply sophisticated mesh processing techniques, and thus incorporate mesh modelling techniques into a

problem of computer vision. Finally, we perform experiments to test the robustness of our algorithm and present a quantitative error analysis for knee joints.

## 2 Twists, kinematic chains and pose estimation

This work is based on a marker-less MoCap system [18, 19]. The human being is represented in terms of free-form surface patches. Joint indices are added to each surface node and the joint positions are assumed. This allows us to generate arbitrary body configurations, steered by joint angles. The corresponding counterparts in the images are 2D silhouettes: These are used to reconstruct 3D ray bundles. A spatial distance constraint is minimized to determine the position and orientation of the surface mesh as well as the joint angles. In this section we give a brief summary of the MoCap system.

### 2.1 Twists

A rigid body motion of a 3D point  $\mathbf{x}$  can be expressed in homogeneous coordinates as

$$X' = (\mathbf{x}', 1)^T = \mathbf{M}\mathbf{X} = \mathbf{M}(\mathbf{x}, 1)^T = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}. \quad (1)$$

The matrix  $\mathbf{R}$  is a  $3 \times 3$  rotation matrix and  $\mathbf{t} \in \mathbb{R}^3$  a translation vector. The set of all matrices of type  $\mathbf{M}$  is called the *Lie Group*  $SE(3)$ . To every Lie group there exists an associated Lie algebra, whose underlying vector space is the tangent space of the Lie group, evaluated at its origin. The Lie algebra associated with  $SE(3)$  is  $se(3) := \{(\mathbf{v}, \boldsymbol{\omega}) | \mathbf{v} \in \mathbb{R}^3, \boldsymbol{\omega} \in so(3)\}$ , with  $so(3) := \{\mathbf{A} \in \mathbb{R}^{3 \times 3} | \mathbf{A} = -\mathbf{A}^T\}$ . Elements of  $so(3)$  and  $se(3)$  can be written as vectors  $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^T$ ,  $\boldsymbol{\xi} = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)^T$  or matrices

$$\hat{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \quad \hat{\boldsymbol{\xi}} = \begin{pmatrix} \hat{\boldsymbol{\omega}} & \mathbf{v} \\ \mathbf{0} & 0 \end{pmatrix}.$$

A twist  $\boldsymbol{\xi}$  can be converted into an element of the Lie group  $\mathbf{M} \in SE(3)$  by computation of its exponential form. That can be done efficiently by using the Rodriguez formula [17].

Note: For varying  $\theta$  the one-parametric Lie-subgroup  $M_\theta = \exp(\theta \hat{\boldsymbol{\xi}})$  yields a screw motion around an axis in space. We will use a degenerate screw (without pitch component) for the model joints.

### 2.2 Kinematic chains

A kinematic chain is modeled as the consecutive evaluation of exponential functions of twists  $\boldsymbol{\xi}_i$  as done in [4]. A point at an end effector that is additionally transformed by a rigid body motion is given by

$$\mathbf{X}'_i = \exp(\theta \hat{\boldsymbol{\xi}}) \cdot (\exp(\theta_1 \hat{\boldsymbol{\xi}}_1) \cdots \exp(\theta_n \hat{\boldsymbol{\xi}}_n)) \cdot \mathbf{X}_i. \quad (2)$$

In the remainder of this paper we will note a pose configuration by the vector  $\chi = (\xi, \theta_1, \dots, \theta_n) = (\xi, \Theta)$  of dimension  $(6 + n)$  consisting of the 6 degrees of freedom for the rigid body motion  $\xi$  and the joint angle vector  $\Theta$ . In our setup, the vector  $\chi$  is unknown and has to be determined from the image data.

### 2.3 Silhouette extraction

In order to find the silhouette of an object in the image, a level set function  $\Phi \in \Omega \mapsto \mathbb{R}$  is employed. It splits the image domain  $\Omega$  into two regions  $\Omega_1$  and  $\Omega_2$  with  $\Phi(\mathbf{x}) > 0$  if  $\mathbf{x} \in \Omega_1$  and  $\Phi(\mathbf{x}) < 0$  if  $\mathbf{x} \in \Omega_2$ . The zero-level line thus marks the boundary between both regions.

For an optimum partitioning, the following energy functional is minimized, which is an extended version of the Chan-Vese model [7]:

$$E(\Phi, p_1, p_2) = - \int_{\Omega} ( H(\Phi(\mathbf{x})) \cdot \log p_1(I(\mathbf{x})) + (1 - H(\Phi(\mathbf{x}))) \cdot \log p_2(I(\mathbf{x})) + \nu \cdot |\nabla H(\Phi(\mathbf{x}))| ) dx \quad (3)$$

with a weighting parameter  $\nu > 0$  and  $H(s)$  being a regularized version of the Heaviside (step) function, e.g. the error function. The probability densities  $p_1$  and  $p_2$  measure the fit of an intensity value  $I(\mathbf{x})$  to the corresponding region. We model these densities by local Gaussian distributions. The partitioning and the probability densities  $p_i$  are estimated according to the expectation-maximization principle.

### 2.4 Registration, Pose estimation

Assuming an extracted image contour and the silhouette of the projected surface mesh, the closest point correspondences between both contours are used to define a set of corresponding 3D lines and 3D points. Then a 3D point-line based pose estimation algorithm for kinematic chains is applied to minimize the spatial distance between both contours: For point based pose estimation each line is modeled as a 3D Plücker line  $L_i = (\mathbf{n}_i, \mathbf{m}_i)$  [1]. For pose estimation the reconstructed Plücker lines are combined with the twist representation for rigid motions: Incidence of the transformed 3D point  $\mathbf{X}_i$  with the 3D ray  $L_i = (\mathbf{n}_i, \mathbf{m}_i)$  can be expressed as

$$(\exp(\theta \hat{\xi}) \mathbf{X}_i)_\pi \times \mathbf{n}_i - \mathbf{m}_i = 0. \quad (4)$$

Since  $\exp(\theta \hat{\xi}) \mathbf{X}_i$  is a 4D vector, the function  $\pi$  denotes the projection of the homogeneous 4D vector to a 3D vector by neglecting the homogeneous component.

For the case of kinematic chains, we exploit the property, that joints are expressed as special twists with no pitch of the form  $\theta_j \hat{\xi}_j$  with known  $\hat{\xi}_j$  (the

location of the rotation axes is part of the model) and unknown joint angle  $\theta_j$ . The constraint equation of an  $i$ th point on a  $j$ th joint has the form

$$(\exp(\theta\hat{\xi}) \exp(\theta_1\hat{\xi}_1) \dots \exp(\theta_j\hat{\xi}_j)\mathbf{X}_i) \times \mathbf{n}_i - \mathbf{m}_i = \mathbf{0}. \quad (5)$$

To minimize for all correspondences in a least squares sense, we optimize

$$\operatorname{argmin}_{\boldsymbol{\chi}} \sum_i \left\| \pi \left( \exp(\hat{\xi}) \prod_{j \in \mathcal{J}(\mathbf{x}_i)} \exp(\theta_j \hat{\xi}_j) \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \right) \times \mathbf{n}_i - \mathbf{m}_i \right\|^2. \quad (6)$$

The function  $\mathcal{J}(\mathbf{x}_i)$  denotes the set of joints that affect the point  $\mathbf{x}_i$ . Linearization of this equation leads to three linear equations with  $6 + j$  unknowns, the six pose parameters and  $j$  joint angles. Collecting enough correspondences yields an over-determined linear system of equations and allows to solve for these unknowns in the least squares sense. Then the Rodriguez formula is applied to reconstruct the group action and the process is iterated for the transformed points until convergence.

## 2.5 The tracking system

Since segmentation and pose estimation can both benefit from each other, it is convenient to couple both problems in a joint optimization problem. To this end, the energy functional for image segmentation in (3) is extended by an additional term that integrates the surface model. Thus, by means of the contour  $\Phi$ , the tracking system in [19] can be described by the following energy functional, which is sought to be minimized:

$$E(\Phi, p_1, p_2, \boldsymbol{\chi}) = - \int_{\Omega} (H(\Phi) \log p_1 + (1 - H(\Phi)) \log p_2 + \nu |\nabla H(\Phi)|) dx + \lambda \cdot \underbrace{\int_{\Omega} (\Phi - \Phi_0(\boldsymbol{\chi}))^2 dx}_{\text{shape error}}. \quad (7)$$

The quadratic error measure in the shape term has been proposed in the context of 2D shape priors, e.g. in [20]. The prior  $\Phi_0 \in \Omega \rightarrow \mathbb{R}$  is assumed to be represented by the signed distance function. This means in our case,  $\Phi_0(\mathbf{x})$  yields the distance of  $x$  to the silhouette of the projected object surface. The influence of the shape prior on the segmentation is steered by the parameter  $\lambda$  (we chose 0.05). Due to the nonlinearity of the optimization problem, we propose an iterative minimization scheme: first the pose parameters  $\boldsymbol{\chi}$  are kept constant while the functional is minimized with respect to the partitioning. Then the contour is kept constant while the pose parameters are estimated to fit the surface mesh to the silhouettes (Section 2.4). A comparable approach for combined segmentation and pose estimation using graph cuts has been presented in [3].

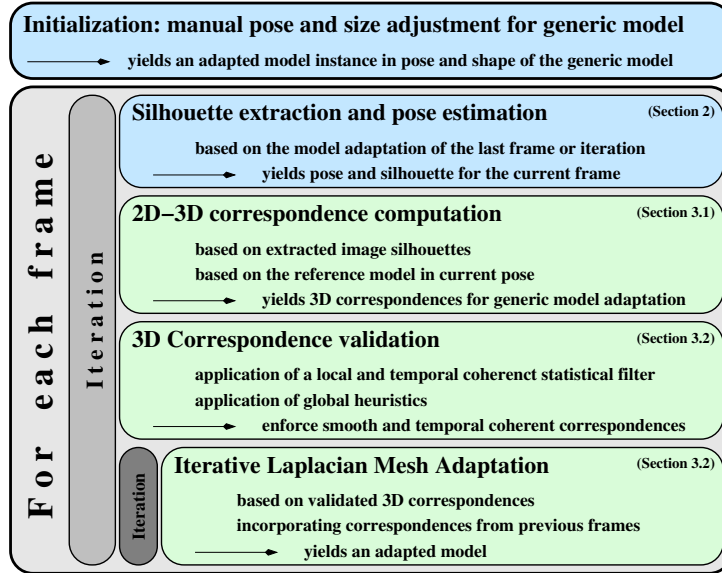
Given the contour  $\Phi$ , the pose estimation method from Section 2.4 minimizes the shape term in (7). Minimizing (7) with respect to the contour  $\Phi$  leads to the gradient descent equation

$$\partial_t \Phi = H'(\Phi) \left( \log \frac{p_1}{p_2} + \nu \operatorname{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) + 2\lambda (\Phi_0(\theta\xi) - \Phi). \quad (8)$$

The total energy is minimized by iterating both minimization procedures. Both iteration steps minimize the distance between  $\Phi$  and  $\Phi_0$ . While the pose estimation method draws  $\Phi_0$  towards  $\Phi$ , thereby respecting the constraint of a rigid motion, in return (8) draws the curve  $\Phi$  towards  $\Phi_0$ , thereby respecting the data in the image.

### 3 Generic Model Adaptation

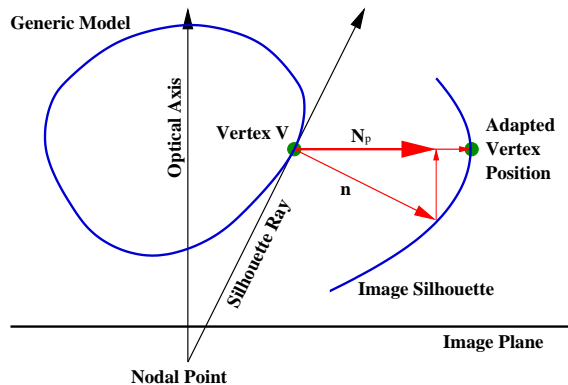
Our adaptation approach extends the tracking loop as sketched in the overview of Figure 2. Given a generic 3D model, the motion tracking algorithm estimates 3D pose and 2D image contours in an iterative process. Each iteration refines the quality for pose- and silhouette estimation.



**Fig. 2.** Model adaptation algorithm incorporated into the basic MoCap system. Green marks the contribution of this paper.

For each new pose estimation the adapted model of the previous iteration is used. Opposite to that, in each frame and iteration the starting point for our adaptation algorithm is the generic 3D mesh model in pose of the current frame.

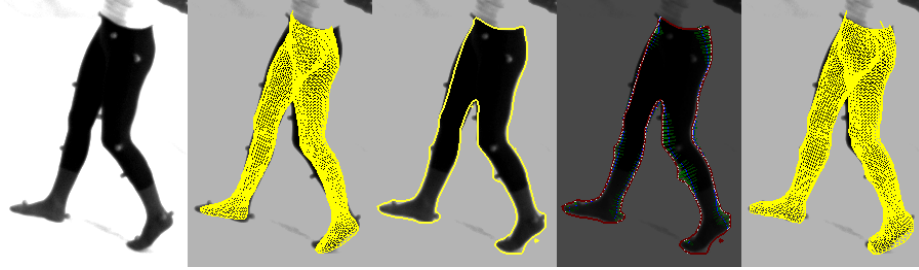
In the further context of this work it is referred to as *reference model* or *reference mesh*. On basis of that reference mesh 3D correspondences are computed for a sparse set of vertices. The correspondences are generated from silhouettes in the image and the reference model along the vertex normals projected to the image planes of the cameras (Figure 3). The reason for projecting to image planes of the cameras is, that, if relying on image contours as seen by a camera, in general it is not possible to state changes parallel to the optical axis (viewing direction).



**Fig. 3.** Sketch for our ICP-algorithm. The algorithm generates a correspondence along normal direction  $n$ . Since, in general silhouettes contain no information about depth, the search is restricted to  $N_p$ , the normal direction projected into the camera plane.

In order to gain smooth and robust adaptation results, the correspondences have to be validated beforehand. That is done by filter heuristics. One filter operates locally and exploits temporal coherence, incorporating all correspondences of previous frames and iterations up to a maximum number (Section 3.2). Relying on statistical means all correspondences, that appear to be noise or outliers, are removed. Furthermore, there are two global heuristics. One removes all correspondences with lengths (i.e. distance between vertex and its displacement position) greater than a given threshold. The other heuristic ensures that only those vertices are involved into mesh adaptation, whose vertex normal is almost perpendicular to the viewing direction. That filter is an important instrument, in order to avoid ill placed correspondences.

Afterwards the correspondences are applied as constraints in a Laplacian mesh adaptation [21, 22] process. The final Laplacian adaptation is designed as an extension of the algorithm as proposed by Stoll et al. [23]. The main extension is to combine all hitherto available correspondences (up to a given number) into one final adaptation routine. Appending the correspondences as constraints to the Laplacian matrix, the solution of the iterative deformation algorithm yields the final adapted mesh model. In all our experiments a number of 15 iterations was sufficient.



**Fig. 4.** Intermediate tracking results for one camera view. *From Left to Right:* Input picture, Reference model in current pose, Extracted Silhouette, 2D Correspondences (blue: ICP starting points, green: search path, white: 2D correspondence point), Final model adaptation

Note: The reference mesh is updated in terms of pose only, but not in terms of shape. If artifacts should occur in the final adaptation result (which may happen in the first few frames), they cannot propagate into the reference mesh. That would lead to non-smooth results or to even more artifacts.

### 3.1 2D-3D Correspondences

The computation of 2D-3D correspondences starts from the generic reference model in pose of the current frame. In order to identify silhouette vertices, the mesh is projected into the camera planes. The silhouette vertices and their projections are stored. Then, for each silhouette vertex an ICP algorithm [26] computes the 2D correspondence to the segmented image contours. The direction of the ICP search is restricted to the projection of the vertex normal to the camera plane. Given the optical axis  $\mathbf{d}$  and normal direction  $\mathbf{N}$  (Figure 3) the projected normal  $\mathbf{N}_{\mathbf{p}}$  is given by

$$\mathbf{N}_{\mathbf{p}} = \mathbf{N} - \langle \mathbf{N}, \mathbf{d} \rangle \cdot \mathbf{N} \quad (9)$$

Thus, the 3D correspondence for a point  $\mathbf{p}$  is found along the Plücker line

$$l : (\mathbf{N}_{\mathbf{p}}, \mathbf{M}), \quad \mathbf{M} = \mathbf{p} \times \mathbf{N}_{\mathbf{p}} \quad (10)$$

If a 2D correspondence point  $(x, y)$  is found, its projection ray is reconstructed. Given the transformation matrix  $[\mathbf{R}, \mathbf{t}] \in \mathbb{R}^{3 \times 4}$  for a camera, the projection ray is represented by the Plücker line  $r : (\mathbf{d}, \mathbf{m})$  with

$$\mathbf{d} = \frac{\mathbf{b} - \mathbf{a}}{\|\mathbf{b} - \mathbf{a}\|}, \quad \mathbf{m} = \mathbf{a} \times \mathbf{d} \quad (11)$$

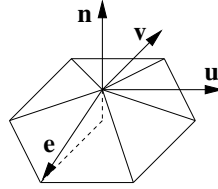
$$\mathbf{a} = -\mathbf{R}^{-1} \cdot \mathbf{t}, \quad \mathbf{b} = \mathbf{R}^{-1} \cdot \left( \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - \mathbf{t} \right) \quad (12)$$

Then, intersecting  $l$  with  $r$ , the final 3D correspondence is computed, i.e. that point on  $l$  that is closest to  $r$ . The intersection  $\mathbf{s}$  is calculated by [1]

$$\mathbf{s} = \frac{\langle \mathbf{N}_{\mathbf{p}}, \mathbf{d} \times \mathbf{m} \rangle - \langle \mathbf{N}_{\mathbf{p}}, \mathbf{d} \rangle \cdot \langle \mathbf{N}_{\mathbf{p}}, \mathbf{d} \times \mathbf{M} \rangle}{(\mathbf{N}_{\mathbf{p}} \times \mathbf{d})^2} \cdot \mathbf{N}_{\mathbf{p}} + (\mathbf{N}_{\mathbf{p}} \times \mathbf{M}) \quad (13)$$

Since correspondences of the ongoing frame are reused in those to come, they are stored. Thus, they have to be transformed into a representation that is invariant to rigid body motion. The global coordinates are transformed to local coordinate systems on vertex basis. Given the vertex normal  $\mathbf{n}$  and the edge to a definite neighbour vertex, we compute coordinate axes  $\mathbf{u}$  and  $\mathbf{v}$ , such that  $\mathbf{u}$ ,  $\mathbf{v}$  and  $\mathbf{n}$  form a local coordinate system (Figure 5).

$$\mathbf{u} = \mathbf{n} \times \mathbf{e}, \quad \mathbf{v} = \mathbf{n} \times \mathbf{u} \quad (14)$$



**Fig. 5.** Coordinate system on a vertex basis: vertex normal  $\mathbf{n}$  and edge  $\mathbf{e}$  to a distinct neighbour vertex are given by the mesh.  $\mathbf{u}$  and  $\mathbf{v}$  are computed such that  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{n}$  are pairwise perpendicular (see Equation (14)).

Given a vertex point  $\mathbf{p}$ , the coordinate axes  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{n}$  and a correspondence point  $\mathbf{c}$ , its new representation is computed by the projections of  $\mathbf{d} := \overrightarrow{\mathbf{p}\mathbf{c}}$  onto  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{n}$ .

$$\tilde{u} = \langle \mathbf{u}, \mathbf{d} \rangle, \quad \tilde{v} = \langle \mathbf{v}, \mathbf{d} \rangle, \quad \tilde{n} = \langle \mathbf{n}, \mathbf{d} \rangle \quad (15)$$

Vice versa  $\mathbf{c}$  is restored by

$$\mathbf{c} = \mathbf{p} + \tilde{u} \cdot \mathbf{u} + \tilde{v} \cdot \mathbf{v} + \tilde{n} \cdot \mathbf{n} \quad (16)$$

### 3.2 Correspondence Analysis

All 3D correspondences are analyzed and filtered by three heuristics:

**(a) Temporal coherent filter over the local variance of correspondences:**

For all 3D correspondences  $\mathbf{c}$  with distance  $d$  to mesh vertex  $\mathbf{p}$  ( $d = \|\overrightarrow{\mathbf{p}\mathbf{c}}\|$ ) the directional distance value  $\tilde{d} = (\tilde{n}/|\tilde{n}|) \cdot d$  ( $\tilde{n}$  from Equation (15)) is computed. Then, within a predefined radius  $r$  (we found 3 cm most suitable) around  $\mathbf{p}$ , the variance  $\sigma^2$  for all correspondences  $c_i$  over the values  $\tilde{d}_i$  is calculated.

$$\sigma^2 = \frac{1}{n-1} \cdot \sum_{i=1}^n (\tilde{d}_i - \bar{d})^2, \quad \bar{d} = \frac{1}{n} \sum_{i=1}^n \tilde{d}_i \quad (17)$$

We also compute

$$\tilde{\sigma}^2 = (\tilde{d}_i - \bar{d})^2 \quad (18)$$

and all correspondences with

$$\tilde{\sigma}^2 > k_1, \quad k_1 \in \mathbb{R} \geq 0 \quad (19)$$

or

$$\tilde{\sigma}^2 > k_2 \cdot \sigma^2, \quad k_2 \in \mathbb{R} \geq 0 \quad (20)$$

are filtered out.  $k_1$  and  $k_2$  are threshold parameters and can be interpreted as follows: If the distance between  $\mathbf{c}$  and  $\mathbf{p}$  differs from the surrounding mean of distances more than  $\sqrt{k_1}$ , or if the ratio of the squared distance  $\tilde{\sigma}^2$  to its approximate *mean*  $\sigma^2$  exceeds  $k_2$ , then  $\mathbf{c}$  is not smooth or reliable enough.

In order to ensure temporal coherent filter results, the list of correspondences used for the filter includes the correspondences of previous frames and iterations up to a maximum number. In our experiments we used a maximum count of 400 sets of correspondence. Older correspondences are deleted.

- (b) **Distance heuristic:** The distance heuristic is a simple global heuristic, removing all correspondences  $\mathbf{c}$  with a distance greater than a given value. This value highly depends on the conformity level of the generic model with the tracked person. In our experiments a maximum distance of 6 cm was sufficient for leg tracking, even for poorly designed generic leg models.
- (c) **Directional heuristic:** Since the correspondences are computed along the direction  $\mathbf{N}_p$  (Figure 3), all vertices with normals parallel to the optical axis would yield no contribution. The more parallel normal and camera direction are, the more unreliable a correspondence is. The angle between vertex normal and optical axis gives a criterion to deal with this issue. If the angle is below a given threshold, the attached correspondence is filtered out. For our needs, angle thresholds of about 75 degrees yield admissible results.

### 3.3 Model Adaptation

Foundation for the final adaptation is a linear variational surface deformation method [2]. We choose a Laplacian Mesh Processing [21, 22] implementation that is based on cotangent weights. Given a reference mesh  $M = (P, E)$  ( $P$  a set of vertices ( $\mathbf{p}_i$ ) and  $E$  a set of edges ( $\mathbf{p}_i, \mathbf{p}_j$ )), the Laplacian scheme encodes the knowledge about structural details of the model  $M$  in terms of differential coordinates that are stored in a vector  $\mathbf{d}_p$ . They are computed by solving  $\mathbf{d} = \mathbf{L} \cdot \mathbf{p}$  (component-wise for  $x$ ,  $y$  and  $z$ ).  $\mathbf{p}$  is a column vector consisting of either  $x$ ,  $y$  or  $z$  coordinates of all points  $\mathbf{p}_i$  and  $\mathbf{L}$  denotes the Laplacian matrix that is constructed from the model  $M$ . The reconstruction  $\bar{\mathbf{p}}$  (again a column vector)

is subject to a number of constraints (3D point correspondences)  $\mathbf{c}_j$  which leads to minimizing a linear least-squares problem of the form

$$\operatorname{argmin}_{\bar{\mathbf{p}}}\{\|\mathbf{L}\bar{\mathbf{p}} - \mathbf{d}_p\|^2 + \|\mathbf{C}\bar{\mathbf{p}} - \mathbf{q}\|^2\} \quad (21)$$

which can be transformed into a system of linear equations

$$(\mathbf{L}^T\mathbf{L} + \mathbf{C}^T\mathbf{C}) \cdot \bar{\mathbf{p}} = \mathbf{L}^T\mathbf{d}_p + \mathbf{C}^T\mathbf{q} \quad (22)$$

Here  $\mathbf{C}$  is a diagonal matrix with non-negative weights  $C_{j,j} = w_j$  for all correspondence constraints.

Since for each frame the tracking- and adaptation solution is computed as refinement result of multiple iterations, the weights are tuned accordingly. Starting with 0.5, they linearly increase up to 1 in the last iteration. At this we obtained good results with 4 iterations for the first frame and 2 for the rest of the sequence.

In order to produce temporal coherent model adaptations, the constraints of previous iterations (as given by the stored 3D point correspondences) are added to the ongoing Laplacian deformation. However, in order to provide the possibility to change the adapted shape over time, the constraint weights are adjusted according to their age, such that they diminish. That is done by multiplying the weights by  $(1 + k)^{-t}$ , where  $t$  is the age and  $k$  a small positive value. Storing constraints for a maximum of 400 iterations, we used  $k = 1/400$ .

A consequence of that approach is, that there may be more than one correspondence that is attached to a vertex. But there are also vertices, without any correspondence attached. Following the idea of Stoll et al. [23] that issue is addressed by exploiting the Laplacian framework for a least-squares approximation for harmonic interpolation [25] over correspondence weights  $w$  and vertex displacements  $\tilde{u}$ ,  $\tilde{v}$  and  $\tilde{n}$ . It is computed by solving the Laplace systems  $\mathbf{L}\mathbf{w} = \mathbf{0}$ ,  $\mathbf{L}\tilde{\mathbf{u}} = \mathbf{0}$ ,  $\mathbf{L}\tilde{\mathbf{v}} = \mathbf{0}$  and  $\mathbf{L}\tilde{\mathbf{n}} = \mathbf{0}$ . Finally, the interpolation result contains as many correspondences as there are vertices in the reference mesh.

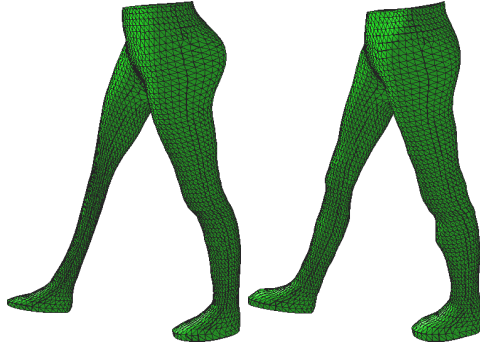
Note: With the exception of vertices without correspondence, the interpolation over all correspondences of past frames is also used for filter (a) in Section 3.2. Independent of the number of previously stored correspondences that smoothes the set of correspondences and limits computational filter costs to a distinct maximum.

In the end, an iterative mesh deformation algorithm yields the adaptation for the ongoing tracking iteration. In iteration  $t$  this algorithm first performs a Laplacian deformation according to tentative constraints. Then, it measures the distance  $l$  between correspondence point and mesh vertex of the tentative adaptation. If exceeding a given maximum distance  $l_{max}$ , the correspondence is excluded from iteration  $t + 1$ . For all remaining correspondences the weights are adjusted by multiplication with  $(1 - l/l_{max})^2$ . In experiments we yield good results performing 15 iterations and using values of 5 to 6 cm for  $l_{max}$ .

## 4 Experiments

We tested our algorithm in several experiments. Playing with the parameter values we found, that in terms of smoothness and robustness the parameters  $r$ ,  $k_1$  and  $k_2$  in Section 3.2 influence the adaptation result most. Suitable values are  $r = 30 \text{ mm}$ ,  $k_1 = 100 \text{ mm}^2$  and  $k_2 = 1$ .

We restrict our experiments to tracking the legs of a person in different motion sequences. The actor wears a tight suit. At distinctive positions there are additional markers attached, sticking out of the legs (see Figure 4). The markers are used in order to compute ground-truth for our motion sequences with a marker-based MoCap system. Thus, secondary objective in our experiments is trying to adapt a shape model that also includes these markers.

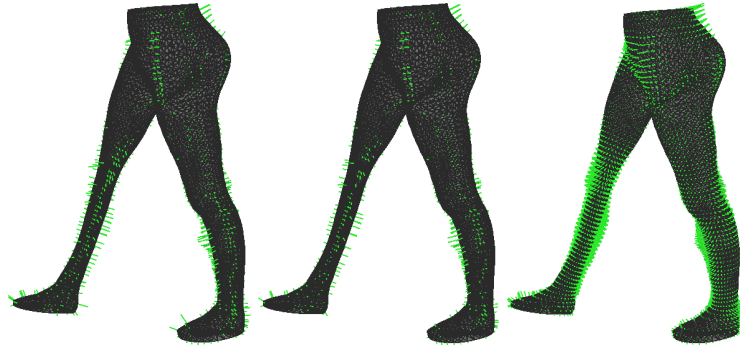


**Fig. 6.** Leg model in pose of the first frame of a walking sequence. *Left:* Degenerated reference model. The right leg is thinned out, and the left half of the backside is blown up. *Right:* Model adaptation after the first iteration.

We test our algorithm in walking and jumping scenes with a degenerated generic leg model. The right leg is thinned out, and the left half of the backside is visibly blown up. Figure 6 shows that model as well as its adaptation after the first iteration of the first frame of a walking sequence.

In order to reliably extract silhouettes, the shape prior  $\lambda$  of equation (7) is adjusted to low values. However, similar to the adaptation result in Figure 4, in both scenarios most markers are too tiny to be reflected within the silhouette. Often, the model adapts the markers at the shinbones only - and in form of smooth bumps.

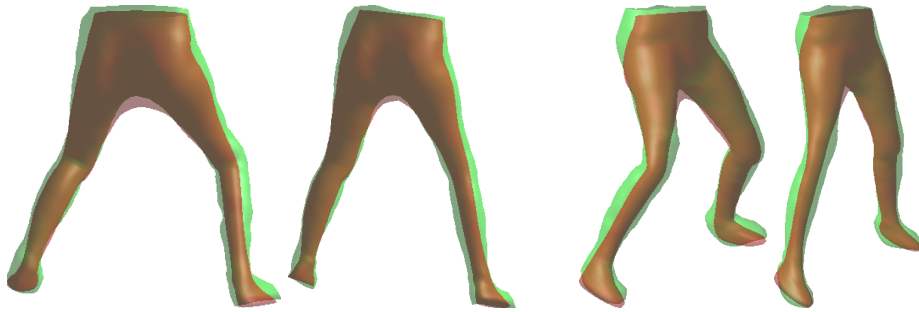
Figure 7 visualizes typical 3D correspondences that are computed. The left image shows the correspondences after applying filters (b) and (c) from Section 3.2. The center image reflects the effects of additionally applying filter (a). The field of correspondences is smoothed. Some visible outliers at the left foot are removed. The right image shows the set of correspondences after harmonic interpolation. They are used for the final adaptation. The thin leg grows to its



**Fig. 7.** 3D correspondences, first frame, first iteration. *Left:* Correspondences after application of filters (b) and (c) from Section 3.2. *Center:* Additional application of filter (a). *Right:* Correspondences after application of filter (a) and harmonic interpolation.

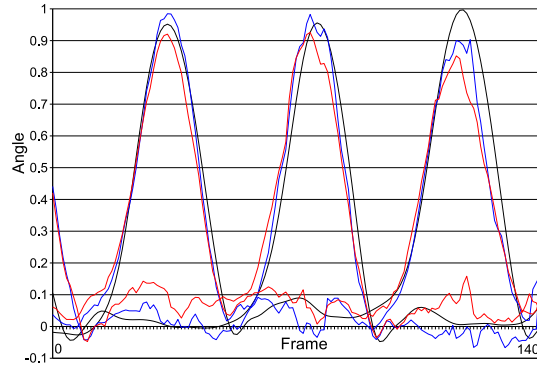
true size, and the thick backside is thinned out. The marker on the left shin-bone results in a smooth bump. Note: Since heuristic (a) implements a filter for temporal coherence, in the first frames it has fewer effects.

Analyzing the effects of adaptation on the generic model, Figure 8 presents model overlays for frames 0 and 10 of the jumping scene. The template model is rendered in red, its adaptation in green. Both are overlaid by addition and 50% alpha channel. Pure green regions show where the template has grown, red regions reflect shrinking.



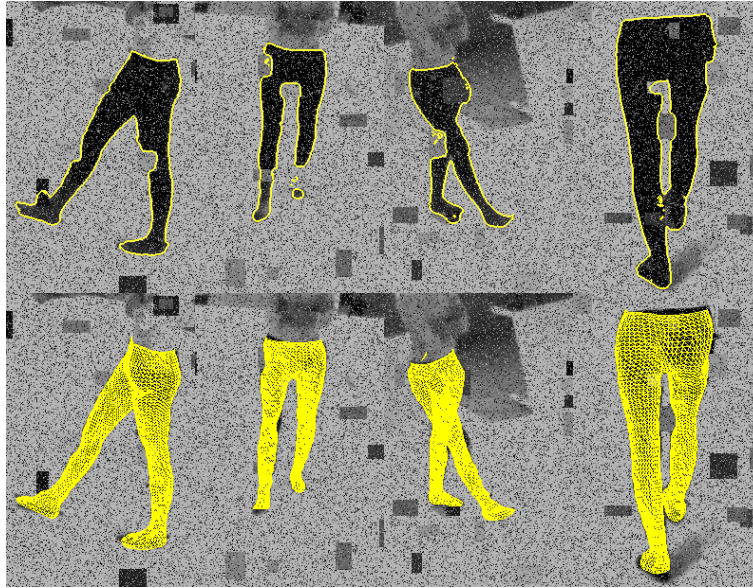
**Fig. 8.** Adaptation analysis: Reference model and its adaptation are overlaid by addition and 50% alpha channel. Red: Reference model, Green: Adapted model. Pure green indicates where the reference model has increased in size, pure red indicates a shrinking.

Figure 9 shows a quantitative error analysis of the walking sequence (frames 0 to 140) for the knee angles. The black lines give the comparison with ground truth as obtained by a marker-based MoCap system. Red represents the angles for the marker-less tracking system without model adaptation. Blue shows the



**Fig. 9.** Quantitative error analysis of the knee angles for a walking sequence (frames 0 to 140). *Black:* Ground truth as obtained by a marker based tracking system. *Red:* Marker-less tracking without model adaptation. *Blue:* Marker-less tracking with model adaptation. *Relative errors:* Blue  $3.4^\circ$ , Red  $4.2^\circ$

results of our approach. The marker-less MoCap system with model adaptation yields results that are closer to ground truth. The relative error for the red curve is  $4.2^\circ$ . The blue curve deviates by  $3.4^\circ$ .



**Fig. 10.** Experiment with uncorrelated noise (25%) and box noise (50 boxes with sizes between  $5 \times 5$  and  $15 \times 15$  pixels). Both rows show a cropped image for each camera view. *Top row:* Silhouette extraction. *Bottom row:* Model adaptation

In more challenging experiments we adjusted the setup by adding noise to the image sequences. Visualizations for tests with uncorrelated noise show adaptation results independent of the noise level that are like those of previous experiments. We also tested with box noise, that is adding boxes of different colors and sizes to the uncorrelated noise. Figure 10 shows silhouette extractions and the adaptation result for the first frame of the walking sequence. We added 25% uncorrelated noise as well as 50 boxes with sizes between  $5 \times 5$  and  $15 \times 15$  pixels. At positions of the boxes the silhouettes are seriously distorted. Though dented, we obtain a relatively smooth model adaptation.

Concentrating on the originally disfigured right leg, Figure 11 shows its adaptations for the frames 0, 5, 10, 15 and 20. The bumps and dents smooth out and finally vanish.



**Fig. 11.** Adaptation result of the right leg for the box noise scenario. Results for the frames 0, 5, 10, 15 and 20.

## 5 Summary

Our approach extends the motion capture process incorporating sophisticated methods for correspondence analysis as well as for mesh processing. It provides a robust method to smoothly adapt any given generic model to the observed shape. That is done using silhouette information only. Since the main goal is to provide smooth and robust adaptation solutions, the algorithm concentrates on low frequency details. High frequency shape details, such as markers on legs, are most likely to be adapted if they are visible in the first frames. Otherwise, they violate our concept for temporal coherence.

We tested the performance of our algorithm on worst case scenarios, limited to tracking legs of a person in different sequences. We applied our algorithm to a clearly sub-optimal generic model and performed a quantitative error analysis for the accuracy of tracking the knee joints. Compared to the non-adaptive approach, our method provides an improvement. We also performed tests with input images that are distorted with intense noise. Our approach yields smooth results.

A weak point, that sometimes occurs in our algorithm, is adapting regions like the feet. The reason for that is, that on one hand shadows on the floor cause

faulty silhouette extractions. On the other hand, all cameras recorded the feet mostly from above, such that it is hard to obtain information about the instep of the feet (see Figures 7 and 8). However, our validation heuristics combined with the smoothing effect of the temporal coherent adaptation approach reduce that issue to a minimum.

## References

1. W. Blaschke. *Kinematik und Quaternionen, Mathematische Monographien*. 4. Deutscher Verlag der Wissenschaften, 1960.
2. M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2007.
3. M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In A. Leonardis, H. Bishof, and A. Prinz, editors, *Proc. 9th European Conference on Computer Vision, Part II*, volume 3952 of *Lecture Notes in Computer Science*, pages 642–655, Graz, May 2006. Springer.
4. C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinetics. *International Journal of Computer Vision*, 56(3):179–194, 2004.
5. A. Bălan, L. Sigal, M. Black, J. Davis, and H. Haussecker. Detailed human shape and pose from images. In *Proc. Computer Vision and Pattern Recognition - CVPR*, June 2007.
6. J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. In *Proc. SIGGRAPH 2003*, pages 569–577, 2003.
7. T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.
8. K.M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63(3):225–245, 2005.
9. E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, USA, June 2007.
10. P. Fua, R. Plänkner, and D. Thalmann. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81(3):285–302, March 2001.
11. L. Herda, R. Urtasun, and P. Fua. Implicit surface joint limits to constrain video-based motion capture. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*, pages 405–418, Prague, May 2004. Springer.
12. N. Magnenat-Thalmann, H. Seo, and F. Cordier. Automatic modeling of virtual humans and body clothing. *Computer Science and Technology*, 19(5):575–584, 2004.
13. I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.
14. T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

15. T.B. Moeslund and E. Granum. A survey of computer vision based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.
16. L. Mündermann, S. Corazza, and T. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In *Proc. Computer Vision and Pattern Recognition - CVPR*, June 2007.
17. R.M. Murray, Z. Li, and S.S. Sastry. *Mathematical Introduction to Robotic Manipulation*. CRC Press, Baton Rouge, 1994.
18. B. Rosenhahn, T. Brox, U. Kersting, A. Smith, J. Gurney, and R. Klette. A system for marker-less motion capture. *Künstliche Intelligenz*, (1):45–51, 2006.
19. B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, 73(3):243–262, September 2006.
20. M. Rousson and N. Paragios. Shape priors for level set representations. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92. Springer, Berlin, 2002.
21. O. Sorkine. Laplacian mesh processing. In *Proc. of Eurographics - STAR Volume*, pages 53–70. Eurographics, 2005.
22. O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, New York, NY, USA, 2004. ACM Press.
23. C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, and H.-P. Seidel. Template deformation for point cloud fitting. In M. Botsch and B. Chen, editors, *Symposium on Point-Based Graphics*, pages 27–35, Boston, USA, 2006. Eurographics.
24. L. You and J. J. Zhang. Fast generation of 3d deformable moving surfaces. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(4):616–615, 2003.
25. R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. In M. Alexa and J. Marks, editors, *Proc. of 26th Annual Eurographics Conference*, volume 24 of *Computer Graphics Forum*, pages 601–609, Dublin, Ireland, 2005. Eurographics, Blackwell.
26. Z. Zhang. Iterative points matching for registration of free form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.