Simultaneous Shape and Pose Adaption of Articulated Models Using Linear Optimization*

Matthias Straka, Stefan Hauswiesner, Matthias Rüther, and Horst Bischof

Institute for Computer Graphics and Vision, Graz University of Technology,
Inffeldgasse 16/II, A-8010 Graz, Austria
{straka,hauswiesner,ruether,bischof}@icg.tugraz.at
http://www.icg.tugraz.at/

Abstract. We propose a novel formulation to express the attachment of a polygonal surface to a skeleton using purely linear terms. This enables to simultaneously adapt the pose and shape of an articulated model in an efficient way. Our work is motivated by the difficulty to constrain a mesh when adapting it to multi-view silhouette images. However, such an adaption is essential when capturing the detailed temporal evolution of skin and clothing of a human actor without markers. While related work is only able to ensure surface consistency during mesh adaption, our coupled optimization of the skeleton creates structural stability and minimizes the sensibility to occlusions and outliers in input images. We demonstrate the benefits of our approach in an extensive evaluation. The skeleton attachment considerably reduces implausible deformations, especially when the number of input views is limited.

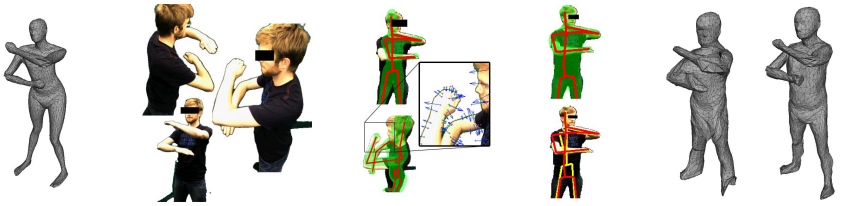
Keywords: Shape Adaption, Pose Estimation, Mesh Editing, Linear Optimization.

1 Introduction

Capturing the shape of a moving non-rigid object from images comprises a variety of challenges: object shape and pose change in every frame and usually only a limited number of synchronized views of the object is available. A typical scenario is to track the temporal evolution of skin and clothing of a human actor without markers in order to record realistic animations. Similarly, taking body measurements without contact and performing motion analysis in sports and medicine often require a purely vision based system. However, the human body is highly articulated and the space of possible shapes is large. Fortunately, most articulated objects are supported by a skeleton which limits the space of possible deformations. This makes it possible to first estimate the coarse skeleton pose and then refine a shape model to represent local details.

* This work was supported by the Austrian Research Promotion Agency (FFG) under the BRIDGE program, project #822702 (NARKISSOS).

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 724–737, 2012.



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

Fig. 1. Our approach adapts the shape and pose of an articulated template mesh (a) to images (b). Distances between vertices and silhouette contours (c) are minimized until the mesh fits tightly to the input silhouettes (d). Our linear formulation allows to optimize shape and skeleton pose simultaneously, which yields more plausible results (f) than optimization of the shape alone (e).

In previous work, the skeleton is only used for pose estimation and initialization of a polygonal mesh such that it has a similar pose. Usually, a subsequent refinement of the mesh surface is decoupled from the skeleton [7,17] because joint optimization can introduce non-linearities. We argue that dropping this association of skeleton and surface makes the adaption process sensible to outliers and can lead to unnatural deformations. Linear blend skinning [3] is the most common method for attaching a mesh to a skeleton. It binds each vertex to one or multiple bones. A mesh in a new pose is obtained by applying bone transformations on each vertex and blending the results. This method is practical for obtaining new vertex positions when bone transformations are given. However, adapting the skeleton from vertex positions is a non-linear problem. Recent work [10,18] has shown that it is possible to attach a skeleton to a mesh using purely linear formulations. Since these methods need to approximate elongated bones by multiple short ones, they are not well suited for the human body.

In this paper, we derive a novel formulation for the deformation of an articulated mesh. It not only allows to optimize for pose and shape jointly but can also be computed efficiently. In contrast to transformation based skinning methods [3,9], our formulation defines the position of each surface vertex relative to the associated bones. The resulting least squares attachment between surface and bones allows to obtain an optimal deformation using linear solvers. Given a roughly initialized mesh, we employ an iterative closest point (ICP) scheme which alternates between a search for correspondences of mesh vertices with image data, and the optimization of vertices and skeleton joints such that the mesh best fits to the given data. In order to improve robustness to unsuitable correspondences, we use a covariance based weighting scheme [12]. It automatically weights point correspondences by analyzing adjacent surface information. Another advantage of our approach is that the size of the skeleton is not assumed to be constant (*i.e.* the length of bones can vary during optimization). For example, this allows to better approximate bending and stretching of the spine of vertebrates using a much simpler model. By introducing quadratic constraints for bone lengths, our method is able to enforce symmetric or fix-sized bones

without increasing the time complexity of the solver. Figure 1 gives a visual overview of our approach.

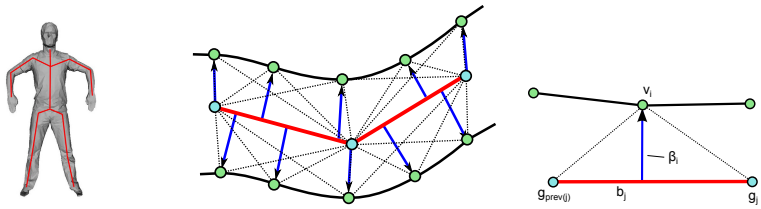
In Section 2, we investigate existing approaches for adapting polygonal meshes to image data. We present our novel formulation for implicit skinning and correspondence weighting in Section 3. In Section 4, we demonstrate how to deform a 2D or 3D mesh to image silhouettes and evaluate the achieved quality and accuracy. We discuss our findings and come to conclusions in Section 5.

2 Related Work

Various authors in the area of computer vision have studied the problem of adapting a polygonal mesh to image data. Previous work can be categorized into two main groups based on whether or not explicit skeletons are used:

Skeleton-less methods: Authors such as Aguiar *et al.* [1] argue that a skeleton limits the application of marker-less motion capture to humanoid models and loose clothing can not be handled realistically. [1] adapt a human body mesh to multi-view images by first deforming a low-resolution volumetric mesh to capture the pose. Then, they refine the detailed shape using a coupled high resolution surface mesh. In contrast, Cagniart *et al.* [5] decompose a mesh into larger surface patches and fit them to 3D point clouds. This increases robustness to noisy data and allows to handle arbitrary objects such as a ball.

Skeleton based methods: These methods encode prior information about the deformable object by using an explicit skeleton. [7] and [17] adapt the shape and pose of a mesh in a two step algorithm which first globally optimizes the skeleton pose and then drops the binding to the skeleton to non-rigidly deform the surface of the mesh. While Vlasic *et al.* [17] propose to optimize the skeleton and polygonal surface independently from each other, Gall *et al.* [7] use mesh vertices rigged to the skeleton to estimate the pose of the model given the image data. In [2], a method to perform pose estimation for the skeleton of a mesh directly with linear blend skinning is presented. A common limitation of above mentioned methods is that the skeleton size must be known in advance. This issue is addressed by Droeschel and Behnke [6] who propose a method to adapt both the pose and some parametric shape parameters of an adaptive body model to image data. Hofmann and Gavrila [8] attempt to optimize both pose and shape of a human model by batch processing a set of automatically selected multi-view frames. Recent work by Taylor *et al.* [16] improves the commonly needed iterative correspondence search between mesh vertices and image data. They train a regression function that can predict correspondence between image data and vertices directly. However, most previously mentioned approaches are based on non-linear optimization and thus they are not computationally efficient for large meshes. A linear method to jointly optimize the skeletal pose and non-rigid shape has not been addressed in the current literature.



(a) Rigged Mesh (b) Implicit Skinning (c) Bone Bindings

Fig. 2. Implicit skinning: Vertices v_i of the mesh (a) are attached to the skeleton (b) by differential coordinates β_i (c)

Our approach is related to the work of [7] and [1]. Instead of using the skeleton only for pose initialization, we integrate it into a linear optimization formulation where it makes the deformation process more robust, yet computationally feasible. Moreover, the simultaneous optimization of pose and shape yields more accurate joint positions than estimating the skeleton pose using a static surface and rotational movements only.

3 Simultaneous Linear Skeleton and Shape Refinement

We formulate the adaption of a polygonal mesh as the minimization of an energy E. This formulation combines a surface smoothness energy E_{skin}, an energy E_{bone} which ensures that vertices are kept attached to their corresponding bones and the energy E_{deform} that allows to deform a mesh to fit image data:

$$E(\mathbf{V}, \mathbf{G}) = \lambda_{\text{skin}} E_{\text{skin}}(\mathbf{V}) + \lambda_{\text{bone}} E_{\text{bone}}(\mathbf{V}, \mathbf{G}) + \lambda_{\text{deform}} E_{\text{deform}}(\mathbf{V}) \qquad (1)$$

where $\mathbf{V} = \{v_i\}, i = 1 \dots |\mathbf{V}|$ represents the vertices of the polygonal mesh $\mathcal{M}$. Our skeleton consists of a set of joint positions $\mathbf{G} = \{g_j\}, j = 1 \dots |\mathbf{G}|$. While vertices of the mesh are connected by faces f_i (3D) or edges (2D), skeleton joints are connected by bones b_j between joint g_j and the parent joint $g_{prev(j)}$ in the tree-like skeleton hierarchy (see Figure 2). The use of explicit point positions makes it possible to create a linear relationship between skin and bones. Each energy can be expressed as a quadratic function and a minimum of (1) is obtained by solving the corresponding unconstrained linear system of equations. The scalars $\lambda > 0$ allow to adjust the influence of each energy term. In the remainder of this section, we describe the energy terms in detail and show how to efficiently solve (1) with additional bone length preserving constraints.

3.1 Laplacian Surface Deformation Energy

We deform the surface of the mesh by modifying selected vertices while keeping the remaining mesh smooth. Laplacian mesh editing [4] is a computationally efficient method that applies the Laplace operator on each surface vertex v_i to

obtain a vector δ_i that is equal to the offset between v_i and the weighted mean of its 1-ring vertex neighborhood $\mathcal{N}_i$:

$$\mathcal{L}_i(\mathbf{V}) = \delta_i = v_i - \sum_{j \in \mathcal{N}_i} w_{ij} v_j \tag{2}$$

where weights $w_{ij} \geq 0$ are obtained using the co-tangent weighting scheme [4]. The squared deformation energy is defined as

$$E_{\text{skin}}(\mathbf{V}) = \sum_{i=1}^{|\mathbf{V}|} \|\mathcal{L}_i(\mathbf{V}) - T_i(\mathbf{V})\hat{\delta}_i\|^2 \tag{3}$$

where $\hat{\delta}_i$ are the delta coordinates of the undeformed mesh. A disadvantage of (2) is that delta coordinates are not rotation and scale invariant. Sorkine *et al.* [13] present how to implicitly approximate a transformation $T_i(\mathbf{V})$ for each vertex v_i that is linear dependent on eventual new vertex positions and therefore allows (3) to be used even in presence of small rotations and scaling[1]. Throughout the paper, we use $\lambda_{\text{skin}} = 1$.

3.2 Skeleton Binding Energy

The skeleton binding energy E_{bone} is responsible for attaching the surface of the articulated mesh to the skeleton. Given an initial configuration between vertices and bones, this energy penalizes a deviation from this configuration during deformation. While [10,18] propose a linear skeleton binding for short bone segments only, our approach handles arbitrary sized bones and is compatible to existing skinning algorithms. A skinning algorithm such as [3] assigns linear skinning weights $\rho_{i,j} \geq 0$ to each vertex v_i which bind it to one or multiple bones b_j ($\sum_j \rho_{i,j} = 1$). We introduce differential bone coordinates β_i which are similar to δ-coordinates in (2). They encode the position of vertex v_i relative to its connected bones b_j, which are defined through joint positions g_j and $g_{prev(j)}$:

$$\mathcal{B}_i(\mathbf{V}, \mathbf{G}) = \beta_i = v_i - \sum_{j=1}^{|G|} \rho_{i,j} \left(\gamma_{i,j}\, g_j + (1 - \gamma_{i,j})\; g_{prev(j)}\right). \tag{4}$$

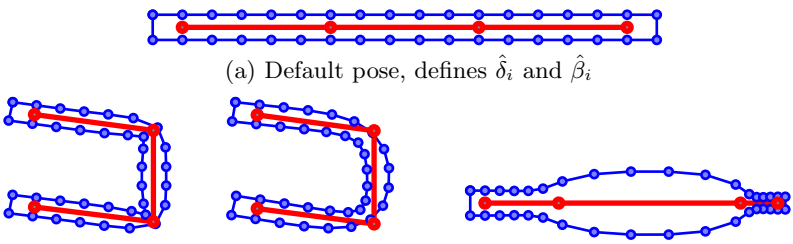
Each $\gamma_{i,j}$ is chosen such that the vector between v_i and $\left(\gamma_{i,j}\, g_j + (1 - \gamma_{i,j})\; g_{prev(j)}\right)$ is orthogonal to bone b_j (see Figure 2c):

$$\gamma_{i,j} = \frac{1}{2} - \frac{\|v_i - g_j\|^2 - \|v_i - g_{prev(j)}\|^2}{2\,\|g_j - g_{prev(j)}\|^2} \tag{5}$$

where $\|.\|^2$ denotes the squared Euclidean distance. The energy E_{bone} penalizes any deviation from this initial attachment:

$$E_{\text{bone}}(\mathbf{V}, \mathbf{G}) = \sum_{i=1}^{|V|} \kappa_i \|\mathcal{B}_i(\mathbf{V}, \mathbf{G}) - \mathbf{T}_i(\mathbf{V}, \mathbf{G})\hat{\beta}_i\|^2 \tag{6}$$

[1] In 2D, $T_i(\mathbf{V})$ can be determined exactly, thus even large rotations and scale changes can be handled without artifacts.



(a) Default pose, defines $\hat{\delta}_i$ and $\hat{\beta}_i$

(b) $\lambda_{\text{bone}} = 2$ (c) $\lambda_{\text{bone}} = 0.1$ (d) Changing bone lengths

Fig. 3. Effects of bending and stretching the skeleton (red) on a 2D mesh (blue). In (b), we show that high values for λ_{bone} cause a stiff surface when bending. A small λ_{bone} (c) produces smoother result due to the skin energy E_{skin}. (d) Changing bone lengths leads to scaling of attached the surface.

where $\hat{\beta}_i$ are the beta-coordinates of the initial mesh. During mesh adaption, this energy ensures that bones and surface vertices will be deformed jointly. For example, joint positions are automatically updated when only surface vertices are deformed, and vice versa. In order to handle small changes in rotation and scale, we implicitly approximate a transformation matrix $\mathbf{T}_i(\mathbf{V}, \mathbf{G})$ for each vertex according to [13]. $\kappa_i \geq 0$ are weights that adjust the strength of the binding between surface and skeleton for each vertex individually. This allows some regions of the mesh to be more rigid than others. A typical scenario is loose clothing, such as a skirt on a human mesh. Stoll *et al.* [14] show how to learn such weights automatically.

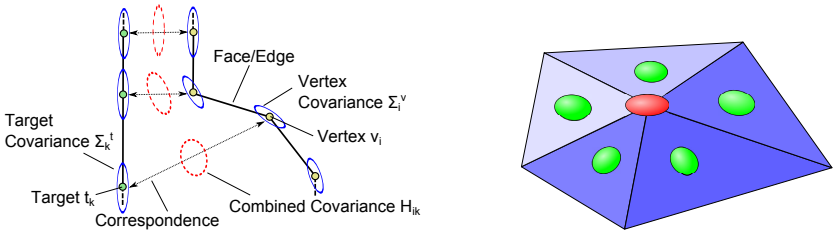
Equation (6) can be used for 2D and 3D meshes and allows natural deformations near joints where vertices are affected by more than one bone. In Figure 3, we show some examples for deforming a 2D mesh. The four skeleton joints are predefined and the new vertex positions are obtained by minimizing $E = E_{\text{skin}}(\mathbf{V}) + \lambda_{\text{bone}} E_{\text{bone}}(\mathbf{V}, \mathbf{G})$.

3.3 Covariance Weighted Mesh Deformation Energy

In an iterative closest point (ICP) scheme, we find correspondences (i, k) between mesh vertices v_i and target points t_k and minimize their squared distance through the deformation energy term E_{deform}:

$$E_{\text{deform}}(\mathbf{V}) = \sum_{(i,k) \in \text{Matches}} \text{dist}^2(v_i, t_k). \tag{7}$$

Related approaches usually minimize the Euclidean distance between model vertices and target points [1,7,17]. This distance measure originates from mesh modeling where deformation is driven by user defined target handles [4,13]. However, when determining correspondences between vertices and target points automatically, a reliable rejection or weighting of implausible matches is required. Instead of using simple surface normal based weighting such as in [17], we extend covariance based correspondence weighting [12] from point clouds to polygonal



(a) Covariance based weighting (b) Surface covariances

Fig. 4. We use covariance matrices for describing the surface around a vertex v_i. The mean of two corresponding covariances yields an automatic weighting $\mathbf{H}_{ik}$.

meshes. [12] is based on the assumption that each vertex v_i can be described by a locally planar neighborhood which is described by a covariance matrix $\mathbf{\Sigma}_i$. It has a small variance in the normal direction of the surface but a high variance on the local surface plane. Such vertex/covariance tuples allow to optimize surface-to-surface distances by minimizing the squared Bhattacharyya distance between v_i and t_k:

$$\mathrm{dist}^2(v_i, t_k|\mathbf{H}_{ik}) = \|v_i - t_i\|^2_{\mathbf{H}_{ik}} = (v_i - t_k)^T \underbrace{\left(\frac{\mathbf{\Sigma}_i^v + \mathbf{\Sigma}_k^t}{2}\right)^{-1}}_{\mathbf{H}_{ik}} (v_i - t_k) \tag{8}$$

where $\mathbf{H}_{ik}$ is the combined correspondence covariance. This distance measure has two key advantages for mesh deformation: First, an anisotropic covariance matrix allows for movement along the surface plane but constrains corresponding vertices to have a low distance in normal direction. Second, the mean of two similarly oriented covariance matrices conserves a high weight in normal direction while differently oriented covariances automatically decrease the strength of the correspondence (see Figure 4a). Note that covariance matrices are symmetric, thus surface points with an opposing normal vector would wrongly yield a strong correspondence. Therefore, we need to discard correspondences with differently oriented surfaces.

For 3D meshes, we calculate the surface covariance $\mathbf{\Sigma}_i^v$ by analyzing the faces adjacent to vertex v_i. First, we assign a covariance $\mathbf{\Sigma}_{f_j}$ to each face:

$$\mathbf{\Sigma}_{f_j} = \mathbf{R}_{f_j} \,\mathrm{diag}(\varepsilon_{\mathrm{cov}}, 1, 1) {\mathbf{R}_{f_j}}^T \tag{9}$$

where the rotation matrix $\mathbf{R}_{f_j}$ rotates the normal of face f_j onto the x-axis and $\varepsilon_{\mathrm{cov}} \ll 1$ defines the variance in normal direction. Then, the covariance $\mathbf{\Sigma}_i^v$ for vertex v_i is obtained as the weighted sum of inverse-covariances of its neighboring faces, as illustrated in Figure 4b:

$$\mathbf{\Sigma}_i^v = \left(\sum_{j \in \mathrm{NF}(v_i)} \alpha_{ij} \left(\mathbf{\Sigma}_{f_j}\right)^{-1} \right)^{-1} \qquad \text{with} \qquad \sum_{j \in \mathrm{NF}(v_i)} \alpha_{ij} = 1 \tag{10}$$

where α_{ij} is a weight proportional to the area of face f_j and $\mathrm{NF}(v_i)$ the list of faces adjacent to vertex v_i. For 2D meshes or silhouette contours, we obtain $\mathbf{\Sigma}_i^v$ via edge covariances weighted by edge lengths.

3.4 Quadratically Constrained Energy Minimization

The quadratic energy minimization from (1) can be efficiently solved using sparse Cholesky matrix decomposition. However, the length of bones is completely unconstrained such that the skeleton can change in size. In situations when there is noisy or occluded image data, the stability of mesh deformation can be increased when the lengths of bones are controlled by quadratic equality constraints. They allow to keep the lengths of bones constant or enforce symmetric bones during shape adaption:

$$\begin{aligned} \min_{\mathbf{V},\mathbf{G}} E(\mathbf{V},\mathbf{G}) = \min_{x} \quad & \frac{1}{2} x^T \mathbf{C} x + b^T x \qquad \text{with} \quad x = \begin{bmatrix} \mathbf{V} \\ \mathbf{G} \end{bmatrix} \\ \text{subject to} \quad & \frac{1}{2} x^T \mathbf{E}_k x = e_k \qquad \text{for} \quad k = 1 \dots K \end{aligned} \tag{11}$$

where the positive definite matrix $\mathbf{C}$ and the vector b encode the least squares equations of (1). $x = [v_1^x, v_1^y, v_1^z, v_2^x, \dots, v_{|V|}^z, g_1^x, g_1^y, \dots, g_{|G|}^z]^T$ contains vertex and joint positions. For example, the length of a single bone can be fixed to e_k using $\|g_j - g_{prev(j)}\|^2 = e_k$, where the squared Euclidean distance operator can be expressed as a symmetric matrix $\mathbf{E}_k$. Similarly, body symmetry (*e.g.* bones with equal length) can be expressed as $\|g_a - g_{prev(a)}\|^2 = \|g_b - g_{prev(b)}\|^2$.

A quadratically constrained quadratic problem (11) cannot be solved using a linear solver directly. Thus, we use the iterative Sequential Quadratic Programming (SQP) algorithm [11]. This algorithm iteratively solves a (sparse) symmetric linear system of equations and therefore increases the time required for solving (11) only by a linear factor.

4 Experimental Evaluation

In this section, we demonstrate that including a skeleton binding energy improves the quality when adapting a polygonal shape to silhouette images. We test our approach on two different types of scenes: adapting the shape and pose of a hand to segmentations of a moving hand in 2D images and adapting a 3D human body model to sequences of multi-view camera images. Hand adaption is particularly suited to show robustness to occlusion and outliers in the input data. The multi-view scenario demonstrates how our skeleton term improves the deformation quality, especially when only a limited number of views is available.

4.1 Optimizing Shape and Pose of a 2D Hand Model

In the first experiment, we optimize the shape and pose of a 2D outline of a hand by adapting it to the segmentation of a real hand. We show how a skeleton

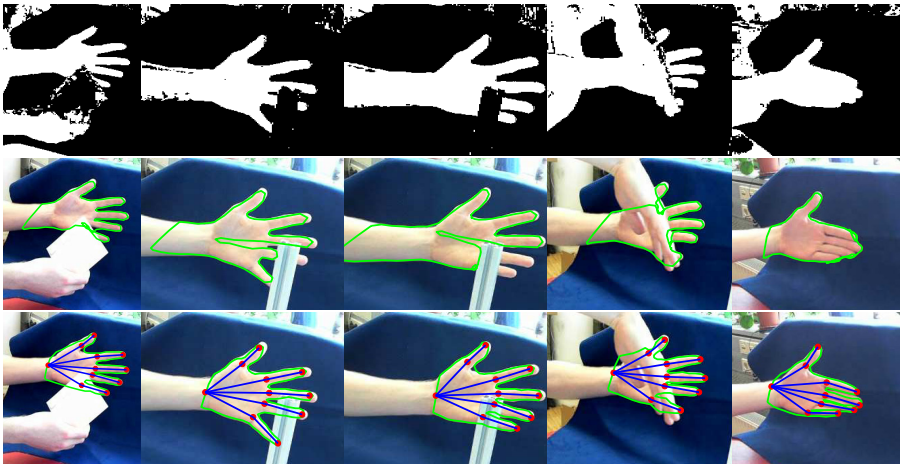


Fig. 5. Adapting a model of a hand to segmented images. Silhouette image (top), adaption without skeleton (middle) and with our skeleton term (bottom).

rigged to the model improves robustness when occluding objects are present in the silhouette image. For comparison, we demonstrate how the same model fails to handle such scenes when no skeletal structure is present.

As a model of the hand, we use a polygonal line with a simple skeleton that allows to pose each finger individually. We obtain skinning weights using the method described in [3] and compute differential coordinates for the line $\hat{\delta}_i$ and skeleton binding $\hat{\beta}_i$ according to (2) and (4). In addition, we set up quadratic constraints $\mathbf{E}_k$ that ensure that the lengths of all bones stay constant during deformation. For evaluation, we recorded multiple sequences of a moving hand, each more than 100 frames long. We manually initialize the positions of the finger tips in the first frame. The remaining bones and the polyline is adapted to the silhouette using correspondences between line vertices and silhouette contours.

In Figure 5, we show some frames of our test sequences (the full videos are provided in supplemental work). We adapt the hand model to the silhouette contours in every frame, using the adapted model of the previous frame as an initialization. The skeleton provides a crucial supportive structure for the polygonal line which helps to maintain a plausible shape and minimizes errors in occluded regions. For example, in Figures 5b and 5c the hand is moved behind a metal bar and a false image contour occurs at the lower finger tips. The skeleton with fixed bone lengths allows to handle this movement correctly, while skeleton-less deformation gets stuck at the contour of the occluder. In Figure 5e, individual fingers are not visible in the silhouette anymore. Nevertheless, the prior information of the skeleton term helps to maintain plausible finger locations. In addition, a skeleton eliminates drifting along the arm.

4.2 Human Shape Adaption from Multi-view Silhouettes

In this experiment, we provide a qualitative and quantitative evaluation of our approach for adapting a 3D human body model to multiple synchronized silhouette images. Similar to [1,7,17], we require a mesh of the person in roughly the same pose. Typically, such a mesh is obtained by a laser scanner or via image based methods. A rough pose can be obtained through pose estimation directly from silhouette images [7,15]. The initialized mesh is then deformed such that its reprojection has a maximum overlap with all silhouette images.

We iteratively align projected rim vertices with their closest silhouette contour in all camera images (8 iterations are usually sufficient). The 2D/3D correspondences are handled by the deformation energy term E_{deform}. In [7], a method is presented that allows to minimize the distance between a 3D vertex $V_i \in \mathbb{R}^3$ and a viewing ray corresponding to an image pixel $t_k \in \mathbb{R}^2$ based on the following linear relationships:

$$\begin{aligned} \left(\mathbf{N}_1^\ell - t_{k,x}\mathbf{N}_3^\ell\right) V_i + \left(T_1^\ell - t_{k,x}T_3^\ell\right) = 0 \\ \left(\mathbf{N}_2^\ell - t_{k,y}\mathbf{N}_3^\ell\right) V_i + \left(T_2^\ell - t_{k,y}T_3^\ell\right) = 0 \end{aligned} \tag{12}$$

where the 3×3 matrix $\mathbf{N}^\ell = \mathbf{K}^\ell\mathbf{R}^\ell$ is calculated from the 3×4 projection matrix $\mathbf{P}^\ell = \mathbf{K}^\ell[\mathbf{R}^\ell|T^\ell]$ of camera ℓ (subscripts of $\mathbf{N}$ denote the respective rows in the matrix). The 3D covariance matrix $\mathbf{\Sigma}_i^V$ corresponding to vertex V_i needs to be rotated into the image coordinate system of camera ℓ, which allows to additively combine it with the 2D contour covariance of pixel t_k:

$$\tilde{\mathbf{H}}_{ik} = \frac{\tilde{\mathbf{\Sigma}}_i^V + \mathbf{\Sigma}_k^t}{2} \qquad \text{with} \qquad \tilde{\mathbf{\Sigma}}_i^V = \begin{bmatrix}\mathbf{R}_1^\ell \\ \mathbf{R}_2^\ell\end{bmatrix} \mathbf{\Sigma}_i^V \begin{bmatrix}\mathbf{R}_1^\ell \\ \mathbf{R}_2^\ell\end{bmatrix}^T \tag{13}$$

where $\tilde{\mathbf{H}}_{ik}$ is the combined correspondence covariance. Plugging (12) and (13) into (8) yields a covariance-weighted squared distance function $\text{dist}^2(V_i, t_k|\tilde{\mathbf{H}}_{ik})$ for 2D/3D correspondences.

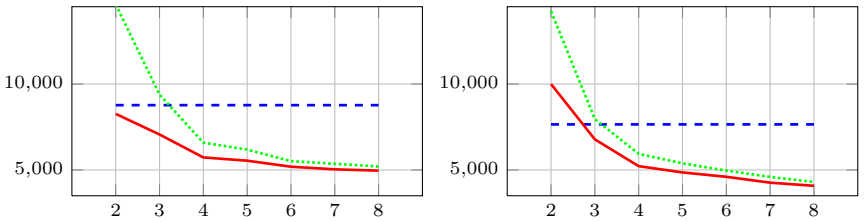
We evaluate our pose and mesh adaption on a public dataset which contains high quality silhouettes of multiple actors recorded by eight 1-megapixel cameras [7]. In every frame, we initialize the actor specific template model using the 3D skeleton pose information provided in this dataset. We make use of our linear skeleton binding energy for shape adaption, but do not use bone length preserving constraints. Our quantitative evaluations are based on the commonly used *pixel overlap error* [1,2,5]. Therefore, we count the number of pixels that are different in the reprojection of the deformed mesh and the input segmentations.

In Table 1, we evaluate the influence of our skeleton term and covariance based correspondence weighting. Almost all scenes benefit from an additional skeleton term, which decreases the silhouette overlap error by 200 pixels on average compared to mesh adaption without skeleton and covariance weighting, which is most similar to the method in [7].

We like to point out that there are larger errors in configurations where a bone term is used without covariance weighting ($\varepsilon_{\text{cov}} = 1$). This can be explained as

Table 1. Effect of covariance weighting ε_{cov} and bone energy when adapting a mesh to multi-view silhouette images. Reported values are the mean silhouette overlap error for the given sequence.

Sequence	# Frames	Not adapted	ε_{cov}=1		ε_{cov}=0.01	
			$\lambda_{\text{bone}} = 0$	$\lambda_{\text{bone}} = 0.1$	$\lambda_{\text{bone}} = 0$	$\lambda_{\text{bone}} = 0.1$
Dance [7]	574	7,600	4,400	4,500	4,300	4,100
Skirt [7]	721	6,900	4,100	4,300	4,300	4,100
Handstand [7]	401	8,800	5,100	5,200	5,200	4,900
Wheel [7]	281	7,200	4,400	4,600	4,300	4,300
Dog [7]	60	4,700	3,300	3,100	3,100	3,100

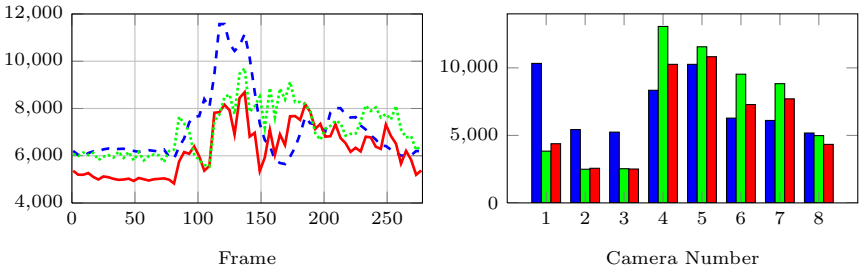


(a) Handstand (401 frames) (b) Dance (574 frames)

Fig. 6. Mean silhouette overlap error in pixels (evaluated on all views) when the mesh is adapted only to the first n views of the given sequence. Legend: not adapted (dashed), without skeleton (dotted) and with skeleton (solid).

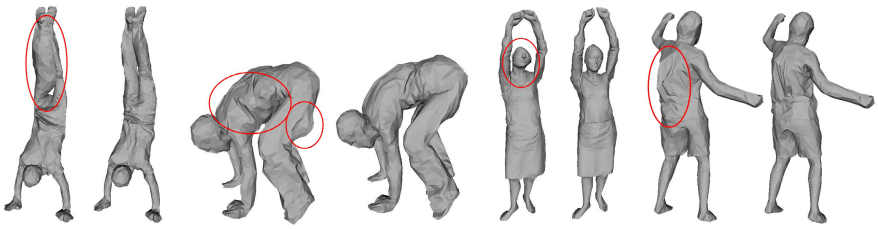
follows: the bone energy competes against the deformation energy to maintain a natural distribution of vertices along the mesh surface while the deformation pulls vertices to their closest silhouette contour. Covariance based weighting enables both energies to be optimized with minimal interference.

So far, the improvement of our method is rather small compared to [7] because the number of camera views (eight) is sufficient for a good adaption with surface-only regularization. The real benefit of our method becomes apparent when fewer input silhouette images are available. In Figure 6, we analyze the silhouette overlap error depending on the number of input views and compare mesh adaption with and without a skeleton term. For reference, we plot the initial error of the not yet adapted mesh, which is independent on the number of views. While there is almost no difference when all eight camera views are used, our bone energy term yields a significantly lower error when only a few views are available. In Figure 7, we take a closer look at the reason for these results. By means of the *Wheel* sequence adapted to the first three camera views, we analyze the mean silhouette overlap at each frame and camera individually. It can be seen that our bone energy consequently yields a lower error in all frames (Figure 7a). While the skeleton term effectively minimizes the errors in views used for adaption, its preference for plausible deformations is honored by a lower error in



(a) Mean error for each frame (b) Mean error for each camera

Fig. 7. Evaluation of the Wheel sequence [7] when only cameras 1-3 are used for mesh adaption. The silhouette overlap error (y-axis) is computed from all views. Legend: not adapted (blue), without skeleton (green) and with skeleton (red).

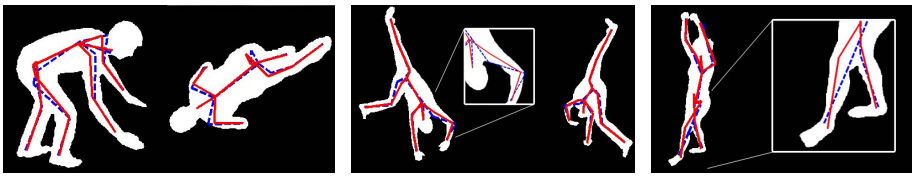


(a) Handstand (b) Handstand (c) Skirt (d) Dance

Fig. 8. Our bone energy term reduces unnatural deformations even when only a few camera views are available (here: 4 views). Deformation without (left) and with skeleton (right).

the remaining views. Adaption without an underlying skeleton simply overfits to the given views and causes unnatural effects visible in remaining views. This can be seen in a qualitative analysis in Figure 8. It is worth noting that the explicit encoding of skeleton joint positions allows to obtain an optimized skeleton pose as a by-product of shape adaption. In Figure 9, we show some frames where our approach significantly improves the locations of skeleton joints over their initial positions.

Finally, we analyze the runtime of our approach when adapting a rather large mesh with 2500 vertices and 15 skeleton joint positions. The overall adaption of the mesh to a single frame in an eight camera setup takes 4 s in an unoptimized Matlab implementation on an Intel i7 CPU, which includes the time for eight iterations of matching rim vertices and silhouette contours, calculating vertex covariances and minimizing the deformation energy. A single minimization of the deformation energy E accounts for about 60 ms when skeleton information is not used. By jointly minimizing the unconstrained bone energy, this time increases to 75 ms. This increase is negligible since energy minimization requires only a fraction of the overall runtime. When solving the quadratically constrained version



(a) Handstand #38 (b) Wheel #120 (c) Skirt #297

Fig. 9. Our simultaneous shape and pose adaption is able to correct an inaccurate initial pose estimate (dashed). Solid lines represent our optimized skeleton.

of (11) with 6 symmetry constraints, the time for a single energy minimization increases to 500 ms.

5 Discussion and Conclusions

We have presented a novel method to bind the surface of a polygonal mesh to an articulated skeleton using a linear formulation. Given an initialized template mesh, we are able to simultaneously adapt both its detailed shape and pose to a silhouette representation of an articulated object such as a human body. This is an improvement over previous methods, which are only able to adapt the mesh surface without making use of skeleton information [1,7,17]. We have shown that a skeleton term in shape optimization is able to increase stability of the deformation process, especially when the number of input views is low. The reason for this improvement is that the skeleton adds a crucial inner structure to the model which penalizes deviations from unnatural deformations. As a consequence, our approach allows to either reduce the number of views in a multi-camera setup or to focus the cameras on different regions of the object to capture more visual details. In order to handle loose clothing, we locally reduce the skeleton binding strength in regions where the surface should not be attached to a skeleton. Our linear skeleton binding can easily be integrated in existing mesh adaption approaches and does not increase the runtime requirements considerably compared to adapting the surface only.

Our approach is not limited to adapt a mesh to 2D silhouette contours. Vertex correspondences and the covariance based weighting scheme can easily be applied to different types of input data, such as 3D point clouds. A limitation shared with other ICP based methods is the dependency on a good initialization of the template mesh. Thus, we can only adapt to local details in the vicinity of the initialized position. By introducing correspondences based on texture information, it is possible to find correspondences across camera views and even sequential frames. Therefore, our hope is that this paper will inspire future work in areas such as skeleton supported mesh tracking.

References

1. Aguiar, E.d., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. In: Proc. of ACM SIGGRAPH (2008)
2. Ballan, L., Cortelazzo, G.M.: Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In: Proc. of 3DPVT (2008)
3. Baran, I., Popović, J.: Automatic rigging and animation of 3D characters. In: Proc. of ACM SIGGRAPH (2007)
4. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE Transactions on Visualization and Computer Graphics 41(1), 213–230 (2008)
5. Cagniart, C., Boyer, E., Ilic, S.: Probabilistic Deformable Surface Tracking from Multiple Videos. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 326–339. Springer, Heidelberg (2010)
6. Droeschel, D., Behnke, S.: 3D body pose estimation using an adaptive person model for articulated ICP. In: Proc. of International Conference on Intelligent Robotics and Applications (2011)
7. Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: Proc. of CVPR (2009)
8. Hofmann, M., Gavrila, D.M.: 3D human model adaption by frame selection and shape-texture optimization. Computer Vision and Image Understanding 115(11), 1559–1570 (2011)
9. Kavan, L., Collins, S., Zara, J., O'Sullivan, C.: Geometric skinning with approximate dual quaternion blending. ACM Transactions on Graphics 27(4), 105 (2008)
10. Li, J., Lu, G.: Skeleton driven animation based on implicit skinning. Computers & Graphics 35(5), 945–954 (2011)
11. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer (2006)
12. Segal, A.V., Haehnel, D., Thrun, S.: Generalized ICP. In: Proc. of Robotics: Science and Systems (2009)
13. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proc. of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 175–184 (2004)
14. Stoll, C., Gall, J., Aguiar, E.d., Thrun, S., Theobalt, C.: Video-based reconstruction of animatable human characters. ACM Transactions on Graphics 29(6) (2010)
15. Straka, M., Hauswiesner, S., Rüther, M., Bischof, H.: Skeletal graph based human pose estimation in real-time. In: Proc. of BMVC (2011)
16. Taylor, J., Shotton, J., Sharp, T., Fitzgibbon, A.: The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In: Proc. of CVPR (2012)
17. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. ACM Transactions on Graphics 27(3) (2008)
18. Zhang, S., Huang, J., Metaxas, D.N.: Robust mesh editing using laplacian coordinates. Graphical Models 73(1), 10–19 (2011)