Character Reconstruction and Animation from Uncalibrated Video

Alexander Hornung[1] Ellen Dekkers[2] Martin Habbecke[2] Markus Gross[1] Leif Kobbelt[2]
[1] ETH Zurich [2]RWTH Aachen University
Technical Report

Abstract

We present a novel method to reconstruct 3D character models from video. The main conceptual contribution is that the reconstruction can be performed from a single uncalibrated video sequence which shows the character in articulated motion. We reduce this generalized problem setting to the easier case of multi-view reconstruction of a rigid scene by applying pose synchronization *of the character between frames. This is enabled by two central technical contributions. First, based on a generic character shape template, a new mesh-based technique for accurate shape tracking is proposed. This method successfully handles the complex occlusions issues, which occur when tracking the motion of an articulated character. Secondly, we show that image-based 3D reconstruction becomes possible by deforming the tracked character shapes as-rigid-as-possible into a common pose using motion capture data. After pose synchronization, several partial reconstructions can be merged in order to create a single, consistent 3D character model. We integrated these components into a simple interactive framework, which allows for straightforward generation and animation of 3D models for a variety of character shapes from uncalibrated monocular video.*

1. Introduction

Recent techniques for image-based 3D character reconstruction have been able to create and animate virtual character models of very high quality. However, most approaches require accurately calibrated systems with multiple synchronized cameras, exact silhouette information, or additional hardware like range sensors. Hence, the corresponding capture setups have become extremely complex and costly. In various application scenarios, *e.g.*, reconstructing historical characters from archived film material or the generation of 3D avatars for home users with a webcam, these techniques are impractical or even inapplicable.

We envision 3D character reconstruction as a simple image processing tool, which allows to reconstruct a virtual model of reasonable quality by simply filming different views of a person with a hand-held camera. In this very general setting, classical approaches to image-based reconstruction fail. One particular reason is that they require multiple views taken at the same time. In monocular video, even small character motion between video frames violates this assumption and renders the reconstruction impossible.

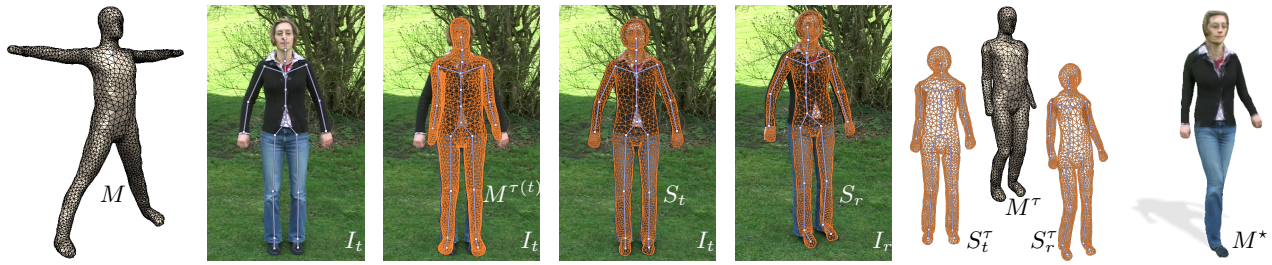
In this paper we show that character reconstruction from a single, uncalibrated video showing a person in articulated motion is nevertheless possible. Our main conceptual contribution is the transformation of this problem into a synchronized multi-view setting by *pose synchronization* of the character. We describe an interactive framework, which takes such a video as input and outputs a 3D model.

This framework is based on two central technical contributions. The first is a novel mesh-based tracking approach, which enables us to accurately track the deformation of the character's shape throughout the video despite of the involved complex occlusions. Secondly, we describe an algorithm which synchronizes the tracked shapes into a common pose using motion capture data in order to compensate for articulated motion. Combined, these two algorithms allow for the generation of consistent 3D reconstructions even in this very general and ill-posed problem setting. As an application we present several different animated 3D character models created from simple, uncalibrated video sequences.

2. Related Work

Research on image-based 3D character reconstruction and animation has primarily focused on controlled acquisition setups with multiple synchronized video streams. One class of techniques reconstructs a character model by computing the visual hull from several silhouettes [25] which can then be refined for a more faithful reconstruction [27, 28] or used, *e.g.* for tracking and pose estimation [21]. These methods require an accurate calibration and a segmentation of the character from the background. Alternatively, a pre-defined human body model can be fitted to the silhouette of a human character [9, 18, 22]. Based on the SCAPE model [3] or other articulated character templates it is even possible to model and animate fine scale shape changes [5, 33]. However, these methods cannot easily be adapted to shapes which differ considerably from the underlying model.

With additional devices like laser-scanners, structured light, or depth cameras it is possible to first acquire a detailed 3D reconstruction offline which can then be deformed



(a) Shape template. (b) Selected joints. (c) Projection. (d) Alignment. (e) Tracking. (f) Synchronization. (g) 3D model.

Figure 1. Overview of the central components of our method. (a) The generic 3D shape template M. (b) From manually selected joint positions in a video frame I_t, the camera projection $\mathbf{P}_t$ and best matching model pose $\tau(t)$ are computed. (c) The template mesh M is deformed according to $\tau(t)$ and projected into I_t. (d) Reference shape S_t after alignment of $M^{\tau(t)}$ to the character's silhouette. (e) Tracking of reference shape S_t to other video frames. (f) Synchronization of the template mesh and the tracked shapes into a common pose. (g) Reconstructed and animated 3D character model $M^\star$.

using multiple segmented video streams [12]. Other approaches reconstruct the pose or shape of an articulated object solely from range-data [10, 26] or provide real-time reconstruction with motion compensation [34]. In combination with 3D motion capture systems, it is also possible to improve the quality of 2D animations from video [14]. All these methods are able to deliver very high quality reconstructions at the cost of complex capturing systems. In contrast, our work addresses the problem of reconstructing a 3D model from a single uncalibrated video only, potentially including articulated motion of the character.

Some techniques reconstruct non-rigid surfaces from monocular input video [17, 30]. However, their specific constraints and the lack of occlusion handling render these methods inapplicable for a practical system for articulated character reconstruction. An alternative is to first estimate the character pose [15] and use this information to facilitate reconstruction. But in order to achieve the accuracy required for 3D reconstruction, further information such as a segmentation is still necessary [1]. Existing methods for camera calibration such as SfM or model-based techniques [16] are not suitable in our setting with a non-static camera and an independently moving character. Since one of our primary aims is to make our method applicable to a variety of character types and input videos, we decided to employ a semi-automatic camera and pose estimation [19].

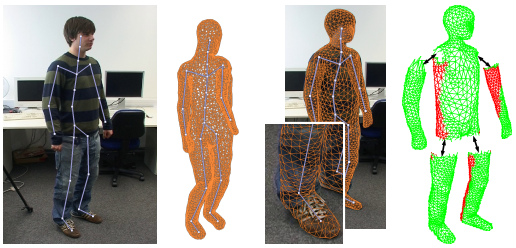
Finally, an important component in image-based 3D reconstruction is a robust approach to accurate pixel correspondence estimation [4, 7, 24]. Optical flow based motion compensation has been used for output view synchronization [35]. However, these methods also do not explicitly handle the above mentioned occlusion problem. Our novel mesh-based tracking approach addresses these issues, without requiring calibrated cameras or specific acquisition setups as other methods [29].

It has already been shown that algorithms for diverse problems in monocular video processing, such as rotoscoping or *rigid* object reconstruction, can be greatly stabilized by integrating very simple user interactions [2, 32]. We pursued a similar design decision and integrated the components of our system into an interactive framework to strike a balance between automation and flexibility. The benefit of such approaches has been recently demonstrated by showing that simple user interaction is sufficient to texture complex 3D models from casual images [31]. In a similar spirit our work aims at reconstructing geometry from casual, uncalibrated input video.

3. Overview

The goal of our method is to reconstruct a 3D model from a single input video of a person. This problem setting is highly ill-posed: Due to articulated motion each frame of the video might show the person in a slightly different pose. Moreover, we do not assume any prior video processing such as camera calibration or image segmentation. We show that, by utilizing a generic 3D character shape template and a database of 3D motion data, a single video of a person can be effectively converted into a temporally synchronized multi-view setup, a process which we refer to as *pose synchronization*. The converted video then allows for an image-based 3D reconstruction of the character shape.

Figure 1 illustrates the central steps of our technique. The first step is to fit a generic shape template M to some input frame I_t of the video. This shape template is a 3D triangle mesh (see Figure 1 (a)) with an embedded skeleton. Given a skeleton pose τ, *e.g.* from a motion database [11], the template can be deformed accordingly $M \rightarrow M^\tau$ using skeleton-driven animation techniques [23]. In order to fit this template model to the character in image I_t we first estimate the approximate character pose $\tau(t)$ and a projection $\mathbf{P}_t$ from the 3D motion data to the 2D image space based on user-selected joint positions (Figure 1 (b)). The deformed template $M^{\tau(t)}$ is then projected into I_t (Figure 1 (c)) and its silhouette vertices are aligned with the character's silhouette (Figure 1 (d)). The resulting *reference shape* S_t, which consists of all projected front-facing triangles, provides an initial mapping from the generic shape template



(a) Joints and estimated pose. (b) Alignment. (c) Layers.

Figure 2. Shape template fitting. (a) Joint positions selected by the user and the computed pose and projection of M. (b) The aligned reference shape S_t. (c) Illustration of the different layers of S_t and the detected occlusions (red regions).

M in 3D to the character in the 2D image I_t (Section 4).

For a 3D reconstruction, dense 2D correspondences between different views of the character are required. This is a challenging problem due to frequent occlusions and the dynamic nature of our input videos. We compute these correspondences by a novel mesh-based approach which is able to track the reference shape S_t through a subsequence of the video $\{I_s, \ldots, I_t, \ldots, I_r\}$ (Figure 1 (e)). Occlusions are resolved by utilizing the depth information of the different layers of front-facing triangles in S_t (see also Figure 2 (c)). The result is a *shape sequence* $\mathcal{S}_t = \{S_s, \ldots, S_t, \ldots, S_r\}$ where corresponding vertices in $\mathcal{S}_t$ have consistent positions on the character (Section 5).

Due to articulated motion of the character between the images, a direct reconstruction from these correspondences is not possible. The main idea of our pose synchronization is to convert the video sequence into a synchronized multi-view setup by eliminating the pose differences. We achieve this by computing an as-rigid-as-possible deformed shape S_j^τ for each tracked shape $S_j \in \mathcal{S}_t$, according to a common pose τ from the motion database (Figure 1 (f)). After this step, a synchronized shape sequence $\mathcal{S}_t^\tau$ corresponds to multiple views of a rigid scene (Section 6).

The one-to-one vertex correspondences between the deformed shapes S_j^τ then allow us to update the 3D shape of the template model. Each shape sequence $\mathcal{S}_t$ contributes a partial update to the template model, *i.e.*, an updated vertex position for every vertex in the respective reference shape S_t. By combining several partial updates from different subsequences of the input video, a consistent 3D model of the filmed character can be created (Figure 1 (g)) and animated with the available motion capture data (Section 7).

4. Shape Template Fitting

As motivated in Section 2, we employ a semi-automatic approach based on a simple user interaction. First, the user selects 2D skeleton joint positions in an image I_t. Examples are shown in Figures 1 (b) and 2 (a). Given these 2D joint positions and a database of 3D motion capture data [11], an approximate Euclidean camera projection $\mathbf{P}_t$ and a best matching pose $\tau(t)$ can be computed by minimizing the re-projection error [19] (Figure 2 (a)).

The reference shape S_t for this image is created by first deforming the generic 3D shape template $M \rightarrow M^{\tau(t)}$ according to pose $\tau(t)$, and then projecting $M^{\tau(t)}$ using the estimated camera model $\mathbf{P}_t$. The triangles of this projected shape serve as an initial guess for the triangles of M actually visible in I_t. However, instead of only the visible parts of the projection $M^{\tau(t)}$, the reference shape S_t stores *all* front-facing triangles with an additional camera-space depth value for each vertex. Hence, S_t is effectively a triangle mesh with connected layers at different depths, corresponding to the limbs of the depicted character (see Figure 2 (c)). This layered representation with depth information is the key property for detecting and resolving occlusions during shape tracking and reconstruction.

The projection only provides an approximate fit of the template M to the character (Figures 1 (c) and 2 (a)). The remaining mismatch is resolved in two steps. First, the projected template S_t is automatically deformed as-rigidly-as-possible [20] using the user-selected 2D joint positions as deformation constraints. Then, the boundary vertices of S_t have to be aligned to the character's silhouette. Existing automatic techniques are error prone in our problem setting without segmentation and with occlusions. Similar to related work [2] we utilize curve-based silhouette editing and let the user match the shape boundaries. The non-boundary vertices of S_t are repositioned automatically in real-time using as-rigid-as-possible shape deformation with the boundary vertices as deformation constraints. This is crucial in order to redistribute the inner triangles within the adapted shape, while preserving their perspectively distorted aspect ratio (see Figure 2 (b)). The result of the shape template fitting is the final reference shape S_t for image I_t. Please see the accompanying video for a demonstration.

5. Shape Tracking

In order to track the character shape we experimented with a variety of standard approaches such as feature tracking [4], correspondence estimation [24], and optical flow [7]. These types of approaches revealed a number of fundamental restrictions. For instance, tracking rectangular windows centered, *e.g.*, at the mesh vertices of S_t, or optical flow has the drawback that it is difficult to handle occlusions and discontinuities at silhouettes of the limbs and body. Techniques for correspondence estimation generally do not provide sufficiently dense matches. These issues become even more severe for the limited character size at standard video resolutions. Moreover, we have to keep track of the complete limbs of a character even if they are partially occluded. We therefore developed a novel mesh-based tracking approach, which exploits the depth information in a layered reference shape S_t in order to resolve occlusions and to keep track of occluded limbs.

Given two successive images I_j and I_{j+1} and a shape S_j

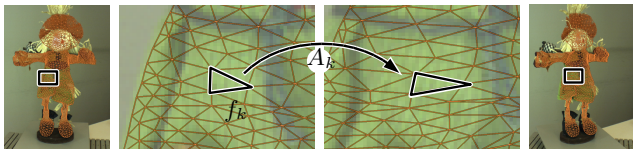


Figure 3. Example of a tracked shape sequence for a more complex model. The close-ups show the deformation of a selected triangle f_k. During the mesh tracking this deformation is approximated by an affine mapping A_k.

in image I_j, our goal is to compute a displacement field attached to the vertices of S_j, *i.e.* a displacement $\mathbf{d}_i$ for each vertex $\mathbf{v}_i$ of S_j. The vertices $\mathbf{v}'_i$ of S_{j+1} then become $\mathbf{v}'_i := \mathbf{v}_i + \mathbf{d}_i$. Each triangle face f_k of S_j, together with the respective transformed face f'_k of S_{j+1}, defines an affine transformation $A_k : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ between the images I_j and I_{j+1} (see Figure 3). We formulate the matching process as a global optimization problem that minimizes the sum of triangle errors. The per-triangle error for each pair (f_k, f'_k) of corresponding triangles is computed by summing over the squared intensity differences of the respective image areas in I_j and I_{j+1}. The desired displacement field is then the minimum of the objective function

$$E_{\text{data}} = \sum_{f_k \in S_j} \sum_{\mathbf{p} \in \Omega_k} \Big(I_j(\mathbf{p}) - I_{j+1}(A_k(\mathbf{p})) \Big)^2, \qquad (1)$$

where $\Omega_k \subset \mathbb{N}^2$ denotes the set of image pixels covered by triangle f_k in image I_j. Coherent tracking and robustness to image noise is ensured by enforcing an additional term

$$E_{\text{smooth}} = \sum_{i \in V(S_j)} \frac{1}{\omega_i} \sum_{j \in N_i} \omega_{i,j} ||\mathbf{d}_i - \mathbf{d}_j||^2, \qquad (2)$$

which imposes rigidity on the tracked mesh. $V(S_j)$ denotes the set of vertices of the shape S_j and N_i denotes the 1-ring neighbors of vertex i. We chose the standard chordal weights $\omega_{i,j} := ||\mathbf{v}_i - \mathbf{v}_j||^{-1}$, $\omega_i := \sum_{j \in N_i} \omega_{i,j}$. The complete objective function then is

$$E = E_{\text{data}} + \lambda E_{\text{smooth}}, \qquad (3)$$

which is minimized using the Levenberg-Marquardt algorithm to determine the vertex displacement field between pairs of successive images.

In order to reliably match large motions of the character we apply a multi-resolution matching approach. The shape meshes for coarse resolution images are generated using a variant of iterative remeshing [6], adjusted to correctly preserve shape boundaries and with an appropriate target edge length. In our experiments we found two resolution levels (*i.e.* the original images and one coarser resolution) to be completely sufficient.

The combination of projected shapes S_t with per-vertex depth information and our mesh-based tracking is the key to resolve the occlusions in the input videos (see Figure 2 (c) and our accompanying video). During the rasterization of the triangles f_k this depth information is taken into account such that each triangle is assigned truly visible pixels only. If a triangle is completely occluded (because, *e.g.*, it lies on the character's torso and is covered by an arm) it is assigned no pixel at all. For the objective function Eq. (3) this has the desirable effect that E_{data} is zero for the respective triangles. The non-zero regularization term E_{smooth} results in a plausible transformation of the occluded parts of the shape S_t. Eventually visible triangles are correctly recognized and included in E_{data}, since the rasterization is performed for every image of a tracked video sequence. Note also that the smoothness term Eq. (2) does not compromise an accurate handling of depth discontinuities between layers and automatically preserves the segmentation of the character and the background.

The choice of λ depends on the orders of magnitude of the energies E_{data} and E_{smooth}. It showed to be rather insensitive to the actual image data, so that we could keep it constantly set to $\lambda = 2$ throughout all of our experiments.

After tracking $S_j \rightarrow S_{j+1}$ the sub-pixel accurate surface matching resulting from this approach effectively corresponds to an exact (stereo) correspondence estimation. The 2D skeleton joint positions are updated by pulling them along the computed displacement field. This tracking loop is iterated both forwards and backwards through the video as long as the reference shape S_t is reasonably consistent with the visible surface areas, *i.e.*, until the viewpoint change with respect to the character in image I_t becomes too large. Termination criteria can be formulated based on the changed aspect ratio of, *e.g.*, boundary triangles or the color mismatch during tracking. We additionally provide the user interactive control when to abort the tracking process, since tracking errors are simple for the user to spot and automatic termination criteria may fail for dynamic video sequences. A viewpoint variation of approximately 30 degrees for a sequence worked well in our experiments.

The result is a sequence $\mathcal{S}_t = \{S_s, \ldots, S_t, \ldots, S_r\}$ which tracks the character's shape from an image I_t through adjacent frames in the input video.

6. Pose Synchronization

Articulated character motion within a tracked image sequence leads to global shape distortions and hence prevents a straightforward reconstruction of the character's 3D shape (see Figure 4 (a)). However, assuming continuous character motion without too large shape changes, a single video sequence can be converted such that it approximates a temporally synchronized multi-view setup by synchronizing all the tracked shapes $S_j \in \mathcal{S}_t$ (see Figure 4 (b)) according to a common 3D skeleton pose τ.

First, camera projections $\mathbf{P}_j$ are computed for the shapes $S_j \in \mathcal{S}_t$ using the procedure described in Section 4. Since the 2D skeleton joints are pulled along with the shapes during the mesh tracking, one generally has to do only minor adjustments to place the joint positions at their approximate locations. The best matching common pose τ for *all* shapes

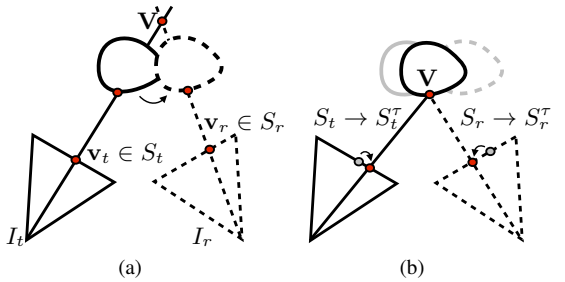


Figure 4. (a) Triangulation of 2D correspondences $\mathbf{v}$ between views of a non-rigid scene results in wrong reconstructions $\mathbf{V}$. (b) Our pose synchronization transforms the correspondences into a rigid setting for a corrected estimate.

$S_j \in \mathcal{S}_t$ can be found by evaluating the combined reprojection error [19]. The actual pose synchronization $S_j \rightarrow S_j^\tau$ is then performed by a 2D as-rigid-as-possible deformation step as in Section 4, with the projected 3D skeleton joints of the common pose as deformation constraints. This step is again motivated by the assumption that the overall visibility of surface triangles remains valid due to continuous character and camera movement, while perspective changes have been captured by the shape tracking. Therefore, the 2D deformation into the common pose is a reasonable approximation of the corresponding 3D pose change (see Figure 5). The result is a synchronized shape sequence $\mathcal{S}_t \rightarrow \mathcal{S}_t^\tau$, which effectively corresponds to the desired multiple synchronized views of a rigid character. This is further illustrated in our accompanying video.

In practice we do the synchronization only for the reference shape S_t and the two shapes S_s and S_r at the boundaries of the tracked video interval. In general the corresponding views have the largest baseline and hence result in the most stable 3D reconstruction.

For a full reconstruction of a character, multiple shape sequences generated from different sections of a video have to be merged into a single, consistent 3D model. When combining two shape sequences with overlapping tracking domains, it is not ensured that a vertex of the template model M ends up at exactly corresponding surface points in the two different sequences. The primary effect of this mismatch is ghosting when merging the texture information during rendering. One solution would be a global optimization of Eq. (3), which involves all shapes in all shape sequences. However, such an approach would be computationally infeasible due to the large number of degrees of freedom and further difficulties such as dynamic appearance changes due to illumination.

We found that an effective solution to this problem is the computation of interpolated 2D vertex positions for each shape similar to the concept of epipole consistent camera weighting [8]. Suppose we have two shapes S_i and S_j from two different shape sequences in the same image. Then some vertices of the template model M are likely to be visible in both shapes S_i and S_j. Let $\mathbf{v}_i$ and $\mathbf{v}_j$ be two corresponding 2D positions of a particular vertex. Then the updated vertex position $\mathbf{v}^\star$ is computed by a weighted contribution $\mathbf{v}^\star = \sum_k \omega_k \mathbf{v}_k$. The weights ω_k are computed based on the number of tracked frames between S_k and the reference shape of the corresponding shape sequence: if S_k is the reference shape (*i.e.*, S_t) of it's respective sequence then $\omega_k = 1$ and all other weights are 0. Otherwise we weight the contribution to the position using $\widetilde{\omega}_k = (|k-t|+1)^{-\beta}$. The final weight is normalized by $\omega_k = \widetilde{\omega}_k / \sum_l \widetilde{\omega}_l$. The exponent β controls the influence of shapes which are distant from the reference image and was set to $\beta = 3$ in our experiments. All vertex positions $\mathbf{v}_i$ are then replaced by the position $\mathbf{v}^\star$.

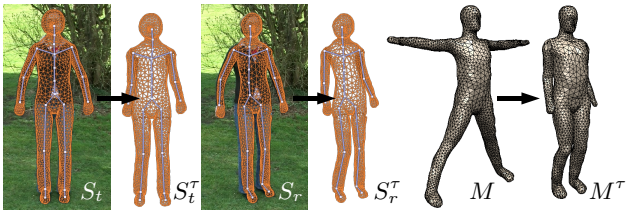


Figure 5. Pose synchronization. S_t and S_r are two tracked shapes of a shape sequence $\mathcal{S}_t$. M is the template model. S_t^τ, S_r^τ and M^τ are the result of the pose synchronization into a common pose τ. Pose changes are visible around the legs and the arms.

Obviously, the mismatch between different shape sequences becomes larger with increasing deviation of the depicted character from the 3D template model M. Hence, it is of relevance mainly for non-human shapes such as the Scarecrow example in Figure 3. However, by initializing the vertex positions in a new shape sequence with tracked vertex positions from a previously generated sequence, it is possible to compute partial reconstructions with pixel-accurate surface points even for such complex models.

7. Reconstruction and Animation

After synchronization, standard multi-view 3D reconstruction is possible using the vertex correspondences between the synchronized shapes S_j^τ in a shape sequence $\mathcal{S}_t^\tau$ and the respective camera projections $\mathbf{P}_j$ (see Figure 4). Each single shape sequence $\mathcal{S}_t^\tau$ allows for a 3D reconstruction of the vertices in its reference shape S_t. In order to generate a single, coherent 3D model, which integrates the information from multiple shape sequences, we compute shape updates for the generic template model M instead of computing independent partial reconstructions.

Hence, for a synchronized shape sequence $\mathcal{S}_t^\tau$, the 3D template is deformed into the same common pose $M \rightarrow M_t^\tau$ (see Figure 5). Then, the 3D positions of its vertices are refined by triangulation [16] of the viewing rays through corresponding 2D vertices of the shapes in $\mathcal{S}_t^\tau$. This is done for each shape sequence separately. The final output model $M^\star$ is generated by deforming all partial reconstructions back into the base pose (Figure 1 (a)) and averaging the positions of all reconstructed vertices. The overall recon-

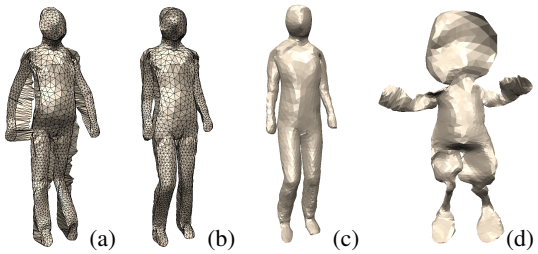


Figure 6. (a) Refinement of the template without pose synchronization of the shapes in Figure 5 results in global distortions. (b) With the synchronization, the template can be properly updated. (c) Full reconstruction from 5 shape sequences. (d) Reconstruction of the Scarecrow model from Figure 3 using 3 shape sequences.

struction is summarized in this algorithm:

foreach *reference image* I_t **do**
  compute S_t and $\mathbf{P}_t$ by fitting template M to I_t;
  track S_t to generate shape sequence
  $\mathcal{S}_t = \{S_s, \ldots, S_t, \ldots, S_r\}$;
foreach *shape sequence* $\mathcal{S}_t$ **do**
  compute camera projections $\mathbf{P}_s$ and $\mathbf{P}_r$;
  synchronize $\mathcal{S}_j \rightarrow \mathcal{S}_j^\tau, j \in \{s,t,r\}$ and $M \rightarrow M_t^\tau$;
  update 3D vertices of M_t^τ using $\mathbf{P}_j, S_j^\tau, j \in \{s,t,r\}$;
foreach *partial reconstruction* M_t^τ **do**
  deform M_t^τ to a common base pose $M_t^\star$;
  use $M_t^\star$ to update the final output model $M^\star$;

Please see also the accompanying video for further illustration of the synchronization and reconstruction step.

The effect of the pose synchronization on the reconstruction is demonstrated in Figure 6: Without synchronization the ray intersection computes wrong depth estimates for the mesh vertices due to the pose differences between the 2D shapes. This causes severe distortions such as the person's left knee bending backwards (Figure 6 (a)). Moreover, the vertices are not properly repositioned with respect to the coordinate system of the template model. This prohibits a proper merging of the different partial reconstructions into a single coherent model. The pose synchronization resolves these problems (Figure 6 (b)) and the vertices of the template can be updated to match the person's shape.

As mentioned in the introduction, one of the main applications we had in mind when developing this method was the possibility to create animated character models from casual, uncalibrated video input recorded with a single camera. A particular feature of our approach is that the reconstructed model $M^\star$ is immediately available for animation, *e.g.*, with different motion capture data, since the original vertex-to-bone weights remain valid. This means that no additional complex rigging has to be performed after reconstruction in order to create an animated 3D model. Although drastic deviations of the reconstructed model from the original template could invalidate the weights, even for the quite complex Scarecrow model (Figure 6 (d)) the original vertex weights are still fine. This is due to the fact that the actual association of vertices to the limbs and torso does not change globally. During rendering we can additionally exploit view-dependent surface textures from the shapes S_j in the input images I_j in order to provide a slightly more dynamic appearance of folds and shadows on the character.

Figure 7. Two frames of an input video and a frame of an animation with new jumping motion overlayed on a different video.

8. Results

We created several animated character models from uncalibrated video or images. The first two examples are based on video recorded with a hand-held camera. In the example in Figure 7 the person was instructed to simply turn around on the spot, which involved a considerable amount of articulated character motion and an unstable background. In situations where the displacement between frames is too large for the mesh-based tracking to converge, our interactive system allows the user to supply a simple but effective hint by translating the affected skeleton joint to its approximate position in the video. But even for this quite challenging input the tracking worked mostly automatically. Figure 6 (a) clearly shows that a reconstruction without considering pose changes results in considerable global deformations and artifacts. Our pose synchronization prior to reconstruction resolves these issues (see Figure 6 (b)).

The full body reconstruction shown in Figures 1 (g), 6 (c) and 7 was created from five shape sequences: three overlapping sequences of the front, and two sequences of the back, each of which covered approximately a viewing angle of 30 degrees (see also our supplemental video). The time required for creating such a 3D character model depends mostly on the number of tracked video frames, since our current implementation is not optimized for speed. For this model each shape sequence consisted of about 40 video frames and took about 30 to 40 minutes to track. The tracking step is the current bottleneck, since all other steps such as the camera and pose estimation and the as-rigid-as-possible shape deformation are a matter of milliseconds to a few seconds. A GPU implementation would allow for real-time tracking and speed up the process considerably.

Similarly, the example in Figure 8 was generated from three shape sequences of the frontal part. Although this does not allow for 360 degree views like for the full body model, it is still possible to animate this model with novel motions involving restricted rotations of the limbs and torso.

Figure 8. A frame of the input video and novel output pictures from an animation with walking motion.

Figure 9. Output animation of the reconstructed Scarecrow model with jumping and waving motion.

In order to demonstrate the flexibility of our system for different character shapes, we applied our method to an input video of a Scarecrow toy, which differs significantly from our generic shape template. Nevertheless, the tracking worked fully automatically for shape sequences consisting of 80 video frames (see Figure 3) and our reconstruction from three such sequences is a faithful reproduction of the tracked surface parts (see Figure 6 (d)). Please note that simple shape approximations like cylindrical elements per limb or purely silhouette-based approaches like visual hulls would not be able to reproduce surface details like the concavities at the head or the trousers. Such errors caused by simpler representations would lead to severe texture registration and blending artifacts during rendering. Frames of a jumping and waving animation created with the resulting 3D model are shown in Figure 9.

We also created a reconstruction and animation of a hand-drawn figure in order to illustrate the possible range of applications for our system. The animation in Figure 10 was generated from three hand-drawn images of the figure from slightly different views and reconstructed from a single shape sequence. Obviously, an accurate correspondence estimation is problematic due to the hand-drawn and hence inconsistent shape and texture between the images. Therefore, tracking was performed at a coarser resolution in order to filter the fine-scale texture inconsistencies. The reconstructed model is quite flat, but still allows to generate 3D animations and rendering effects.

Please see the supplemental video for the full animations and additional material. All animations in the video were generated from a publicly available motion database [11]. Please note that some motion artifacts like foot skating stem from this data. Our resulting models are not restricted to this particular dataset but can be animated with any method suitable for skeleton-driven animation [23].

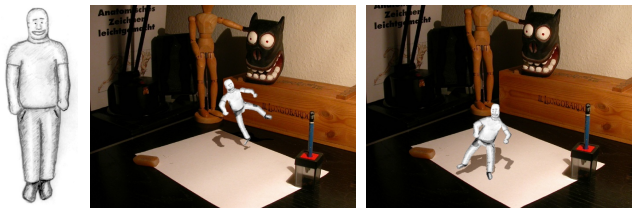


Figure 10. One of three input images and frames of a walking and jumping animation with the reconstructed 3D model.

Limitations. A limitation that we would like to address in future work is that the performance of our system for different input shapes and motions is rather difficult to quantify. As shown in our results, it works reasonably well for pose changes occurring for, *e.g.*, a person turning on the spot, or for challenging shapes like the Scarecrow. However, a reconstruction of a running person is likely to fail due to the considerable pose changes and occlusions between video frames. Even with a high-speed camera, it would be difficult to cover a sufficiently large baseline to extract 3D shape information. Extending this work for general performance capture would be an interesting area of future work.

Another challenge is the reconstruction of surface details. As shown in Figure 6 (d), our method captures more geometry detail than would be possible using simpler approaches such as visual hulls. But it fails to reconstruct the nose of the full body model properly (Figure 6 (c)). A detailed comparison of the reconstruction quality with existing high quality techniques for controlled studio setups (*e.g.*, [12]) seems difficult since our problem setting involves unsegmented, monocular video and independent motion of the background and the character, where those previous works are not applicable. For higher video resolutions, one could integrate dynamic tesselation and refinement techniques of the template model to reconstruct character models with a more detailed surface. With future improvements in the fields of pose detection and segmentation of video we also believe that some manual steps such as the initial shape fitting process could be automated.

Since we extract texture from video, the texture quality of methods based on single, higher resolution images such as [31] is generally better. However, this method cannot reconstruct the geometry of a character and hence requires closely matching 3D models. Our texture quality could be improved using, *e.g.* Floating Textures [13]. Finally, the consideration of shape changes which cannot be explained by articulated motion represents an interesting direction for future work.

9. Conclusion

We presented a novel, semi-automatic approach to reconstruct and animate 3D character models from uncalibrated, monocular video. This very general setting violates

fundamental requirements of existing work on image-based character reconstruction. Our two main technical contributions are a novel model-based tracking approach and an algorithm for pose synchronization to compensate for articulated character motion. We demonstrated that it is possible to produce reasonable reconstruction and animation results for a range of different character types.

Due to the ill-posed problem setting there obviously is a certain trade-off between flexibility regarding the system requirements and the visual quality of the reconstructions. However, we did not intend to compete with the very high quality possible with more complex and constrained capture systems. We believe that our system nicely complements existing work and provides a first basis for making 3D character reconstruction from general video a simple image processing task. This opens up entirely new opportunities and applications such as, for example, the reconstruction of historical characters from film archives or the creation of realistic 3D avatars for home users.

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE PAMI*, 28(1):44–58, 2006. 2

[2] A. Agarwala, A. Hertzmann, D. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM TOG*, 23(3):584–591, 2004. 2, 3

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ACM TOG*, 24(3):408–416, 2005. 1

[4] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004. 2, 3

[5] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *IEEE CVPR*, 2007. 1

[6] M. Botsch and L. Kobbelt. A remeshing approach to multiresolution modeling. In *SGP*, pages 189–196, 2004. 4

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004. 2, 3

[8] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH '01*, pages 425–432, 2001. 5

[9] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM TOG*, 22(3), 2003. 1

[10] W. Chang and M. Zwicker. Range scan registration using reduced deformable models. *CGF*, 28(2):447–456, 2009. 2

[11] CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/, April 2009. 2, 3, 7

[12] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM TOG*, 27(3), 2008. 2, 7

[13] M. Eisemann, B. D. Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating Textures. *CGF*, 27(2):409–418, 4 2008. 7

[14] M. Flagg, A. Nakazawa, Q. Zhang, S. B. Kang, Y. K. Ryu, I. Essa, and J. M. Rehg. Human video textures. In *SI3D*, 2009. 2

[15] S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, and L. J. Van Gool. Articulated multi-body tracking under egomotion. In *ECCV*, 2008. 2

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003. 2, 5

[17] C. Hernández, G. Vogiatzis, G. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *IEEE ICCV*, 2007. 2

[18] A. Hilton, D. Beresford, T. Gentils, R. Smith, W. Sun, and J. Illingworth. Whole-body modelling of people from multiview images to populate virtual worlds. *The Visual Computer*, 16(7):411–436, 2000. 1

[19] A. Hornung, E. Dekkers, and L. Kobbelt. Character animation from 2D pictures and 3D motion data. *ACM TOG*, 26(1), 2007. 2, 3, 5

[20] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM TOG*, 24(3):1134–1141, 2005. 3

[21] R. Kehl, M. Bray, and L. J. Van Gool. Full body tracking from multiple views using stochastic sampling. In *IEEE CVPR*, volume 2, pages 129–136, 2005. 1

[22] W.-S. Lee, J. Gu, and N. Magnenat-Thalmann. Generating animatable 3D virtual humans from photographs. *CGF*, 19(3), 2000. 1

[23] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00*, pages 165–172, 2000. 2, 7

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2, 3

[25] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *SIGGRAPH '00*, pages 369–374, 2000. 1

[26] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller. Object pose detection in range scan data. *IEEE CVPR*, 02:2445–2452, 2006. 2

[27] P. Sand, L. McMillan, and J. Popović. Continuous capture of skin deformation. *ACM TOG*, 22(3):578–586, 2003. 1

[28] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE CG&A*, 27(3):21–31, 2007. 1

[29] S. Sugimoto and M. Okutomi. A direct and efficient method for piecewise-planar surface reconstruction from stereo images. In *IEEE CVPR*, 2007. 2

[30] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. In *ECCV*, volume 2, pages 299–312, 2004. 2

[31] Y. Tzur and A. Tal. Flexistickers: photogrammetric texture mapping using casual images. *ACM TOG*, 28(3), 2009. 2, 7

[32] A. van den Hengel, A. R. Dick, T. Thormählen, B. Ward, and P. H. S. Torr. Videotrace: rapid interactive scene modelling from video. *ACM TOG*, 26(3):86, 2007. 2

[33] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM TOG*, 27(3), 2008. 1

[34] T. Weise, B. Leibe, and L. V. Gool. Fast 3d scanning with automatic motion compensation. In *IEEE CVPR*, 2007. 2

[35] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. E. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM TOG*, 24(3):756–764, 2005. 2