HUMAN SHAPE AND MOTION RECOVERY USING ANIMATION MODELS

P. Fua, L. Herda, R. Plänkers, and R. Boulic
Computer Graphics Lab (LIG), EPFL
CH-1015 Lausanne, Switzerland
{Pascal.Fua,Ralf.Plaenkers,Lorna.Herda,Ronan.Boulic}@epfl.ch

ABSTRACT

Deriving human body shape and motion from optical or magnetic motion-capture data is an inherently difficult task. The body is very complex and the data is rarely error-free and often incomplete. The task becomes even more difficult if one attempts to use video-data instead, because it is much noisier.

In the last few years, we have developed techniques that use sophisticated human animation models to fit such noisy and incomplete data. It is acquired using a variety of devices, ranging from sophisticated optical motion-capture systems to ordinary video-cameras. We use facial and body animation models, not only to represent the data, but also to guide the fitting process, thereby substantially improving performance.

1 INTRODUCTION

In this paper, we show that we can effectively use sophisticated human animation models to fit noisy and incomplete data acquired using a variety of methods. So far, these methods include optical motion capture systems, calibrated sets of images or uncalibrated video-sequences. In all cases, the models are used throughout the tracking and fitting processes to increase robustness. As time goes by, we intend to extend this to an even larger set of acquisition devices.

Optical Motion Capture: It has proved an extremely effective means to replicate human movements. It has been successfully used to produce feature-length films such as "Titanic" that features hundreds of digital passengers with such level of realism that they are indistinguishable from real actors. The most critical element in the creation of digital humans was the replication of human motion: "No other aspect was as apt to make or break the illusion."(Titanic Special Reprint, 1997) Optical motion capture offers a very effective solution to this problem and provides an impressive ability to replicate gestures. Strolling adults, children at play and other lifelike activities have been recreated in this manner. The issues are slightly different for game-oriented motion capture. Capturing subtleties is less important because games focus more on big and broad movements. What matters more is the robustness of the reconstruction process and the amount of human intervention that is required.

In this last respect, commercially available motion capture systems are still far from perfect. Even with a highly professional system, there are many instances where crucial markers are occluded or when the algorithm confuses the trajectory of one marker with that of another. This requires much editing work on the part of the animator before the virtual characters are ready for their screen debuts. To remedy this weakness, we have proposed the use of a sophisticated anatomic human model to increase the method's reliability (Herda et al., 2000).

Video-Based Modeling: The use of markers also tends to make such systems cumbersome. Videogrammetry is therefore an attractive alternative: It uses a cheap sensor and allows not only "markerless" tracking but also precise body-shape modeling. In this work, we combine stereo-data and body outlines. These two sources of information are complementary: The former works best when a body part faces two or more of the cameras but becomes unreliable where the surface slants away, which is precisely where silhouettes can be used.

However, image-based data often is noisy and incomplete. Again, we use the animation models, not only to represent the data, but also to guide the fitting process, thereby substantially improving performance for both face and body modeling. Given ordinary uncalibrated video sequences of heads, we can robustly register the images and produce high-quality realistic models that can then be animated (Fua, 2000). The required manual intervention reduces to supplying the location of 5 key 2–D feature points in one image. For bodies, we recover complex 3–D motions by fitting our articulated 3–D models to the image data (D'Apuzzo et al., 1999, Plänkers et al., 1999).

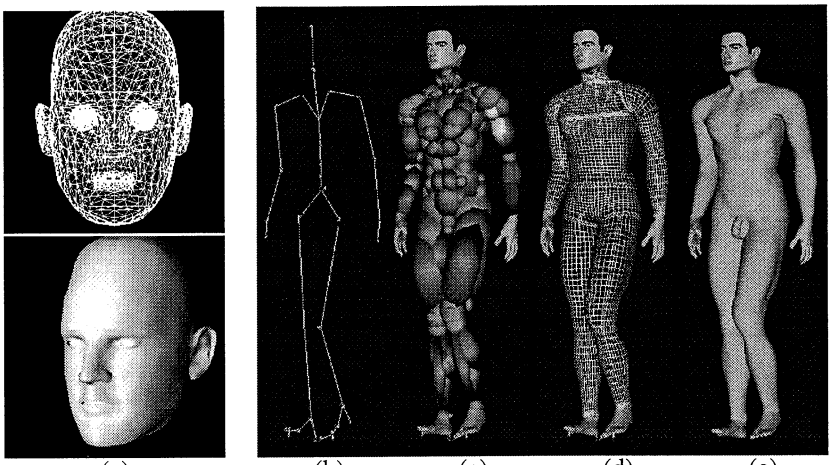


Figure 1: The layered human model: (a) Model used to animate heads, shown as a wireframe at the top and as a shaded surface at the bottom. (b) Skeleton. (c) Ellipsoidal metaballs used to simulate muscles and fat tissue. (d) Polygonal surface representation of the skin. (e) Shaded rendering.

Using Models: Thus, for both optical and video-based motion capture, we have developed robust model-based fitting techniques to overcome ambiguities inherent to the raw data. To be suitable for animation, models tend to have many degrees of freedom and our algorithms are designed to handle them. They can also deal with ambiguous, noisy and heterogeneous information sources, such as optical markers, stereo, silhouettes and 2–D feature locations.

In the remainder of this paper, we first introduce the animation models we use and their specificities. We then discuss our approach to skeleton-based motion capture using optical markers. Last, we present the techniques we have developed for video-based modeling of faces and bodies.

2 ANIMATION MODELS

The modeling and animation of Virtual Humans has traditionally been decomposed into two subproblems: facial animation and body animation.

In both cases, muscular deformations must be taken into account, but their roles and importance differ. Facial animation primarily involves deformations due to muscular activity. Body animation, on the other hand, is a matter of modeling a hierarchical skeleton with deformable primitives attached to it so as to simulate soft tissues. It is possible to model bodies in motion while ignoring muscular deformations, whereas to do so for facial animation is highly unrealistic.

For heads, we use the facial animation model that has been developed at University of Geneva and EPFL (Kalra et al., 1992) and is depicted by Figure 1(a). It can produce the different facial expressions arising from speech and emotions. To simulate muscle actions, we use Rational Free Form Deformations (RFFD) because they are simple, easy to use, intuitive and computationally inexpensive (Kalra et al., 1992). The muscle design uses a region based approach: Regions of interest are defined and associated with a muscle made of several RFFDs. Deformations are obtained by actuating those muscles to stretch, squash, expand and compress the inside facial geometry. For complex expressions, the model may be used to simultaneously render the deformations of various parts of the face.

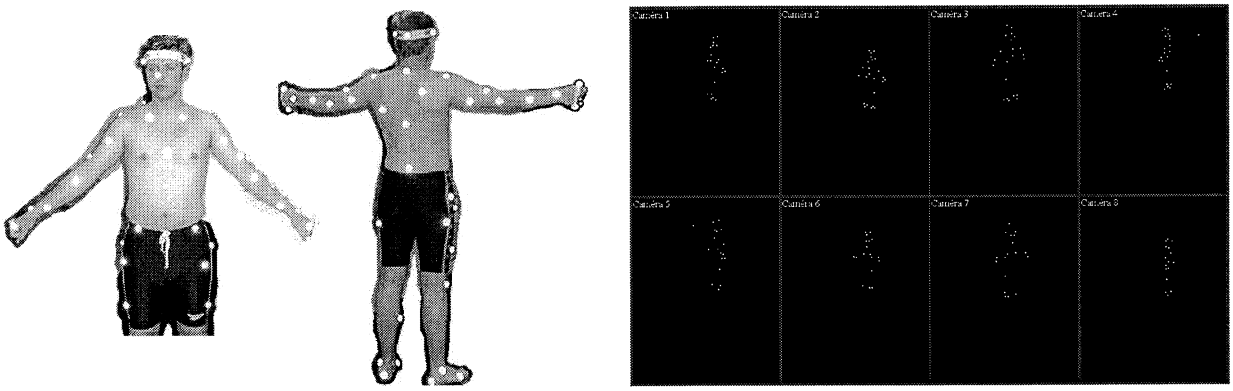
Our body model (Thalmann et al., 1996) is depicted by Figure 1(b,c,d,e). It incorporates a highly effective multi-layered approach for constructing and animating realistic human bodies. Ellipsoidal metaballs are used to simulate the overall behavior of bone, muscle, and fat tissue; they are attached to the skeleton and arranged in an anatomically-based approximation. Skin construction is a three step process: First, the implicit surface resulting from the combination of the metaballs influence is automatically sampled along cross-sections (Shen and Thalmann, 1995, Thalmann et al., 1996). Second, the sampled points become control points of a B-spline patch for each body part (limbs, trunk, pelvis, neck). Third, a polygonal surface representation is constructed by tessellating those B-spline patches for seamless joining of different skin pieces and final rendering. This simple and intuitive method combines the advantages of implicit, parametric and polygonal surface representation, producing very realistic and robust body deformations.

3 SKELETON-BASED MOTION CAPTURE

Our goal is to increase the reliability of an optical motion capture system by taking into account a precise description of the skeleton's mobility and an approximated envelope (Herda et al., 2000). It allows us to accurately predict the 3–D

location and visibility of markers, thus significantly increasing the robustness of the marker tracking and assignment, and drastically reducing—or even eliminating—the need for human intervention during the 3–D reconstruction process.

In contrast to commercially available approaches to motion capture such as the ones proposed by Elite^{tm} and VICON^{tm}, we do not treat 3–D marker reconstruction independently from motion recovery. Instead we combine these two processes and use prediction techniques to resolve ambiguities. For example, we can predict whether or not a marker is expected to be occluded by the body in one or more images and take this knowledge into account for reconstruction purposes. When a marker cannot be reconstructed with certainty from its image projections, we use the expected position of the skeleton to identify the marker and disambiguate its 3–D location. This is helpful when it is only seen by a small number of cameras. In our approach, the performer's skeleton motion is a byproduct of the reconstruction process.



(a) (b)

Figure 2: Input Data: (a) The performer wears markers and is imaged by eight infrared cameras. (b) For each camera, the Elite^{tm} system returns a 2–D location for each visible marker.

We have used as input the 2–D camera data and calibration parameters provided by an Elite^{tm} optical motion capture system (Ferrigno and Pedotti, 1985). More precisely, as shown in Figure 2(b), we are given sets of 2–D point locations, one for each marker and each camera that sees it, and a projection matrix for each camera.

To extract a 3–D animation of a skeleton from a variety of movements performed by the same actor wearing the same markers, we first derive a skeleton-and-marker model, that is a skeleton scaled to the actor's body proportions and an estimate of the markers' locations with respect to the joints. To achieve this result, the actor is asked to perform a "Gym motion." It is a sequence of simple movements that involve all the major body joints. We can then use this calibrated skeleton for further motion capture sessions of more complex motions.

3.1 Acquiring the Skeleton and Marker Model

During the calibration phase, our goal is to scale the bones of the generic skeleton of Figure 3(a) so that it conforms to the performer's anatomy and to model the marker's locations with respect to the joints. The complete skeleton, excluding detailed hands and feet, had 69 degrees of freedom (33 joints), plus six position parameters in 3–D space. The end result is a skeleton-and-marker model such as the one shown in Figure 3(b). In this work, we use a very simple marker model: The markers are attached to specific joints and are constrained to remain on a sphere centered around that joint.

The skeleton-and-marker model is computed using least-squares minimization. As this is a non linear process, the system goes through three successive adjustment steps so as to move closer and closer to the solution at an acceptable cost while avoiding local minima. These steps are described below.

3.1.1 3–D marker reconstruction As the gym motion is an especially simple routine highlighting the major joints motions, the 3–D location of the markers can be automatically and reliably reconstructed without knowledge of the skeleton for 200 to 300 frames at a time. In practice, we partition the gym motion into independent sequences, each one involving only the motion of one limb or body part at a time. We then perform 3–D reconstruction and tracking for each one independently. If necessary, the user can reattach some markers to specific body parts if they become lost.

3–D markers are reconstructed from the 2–D data using stereo triangulation (Faugeras and Robert, 1996). In our examples, we use eight cameras. We first perform pairwise reconstruction. For each non-ambiguous stereo match, that is when there is only one possible candidate, we compute the corresponding 3–D coordinates on the basis of the 2–D coordinates. These 3–D coordinates are then re-projected onto the remaining six camera views, in order to determine the entire set of 2–D coordinates potentially associated with this one 3–D marker. We assume that a 3–D marker is correctly reconstructed if it re-projects into at least one other camera view, thus making a total of at least three camera views. We will say that

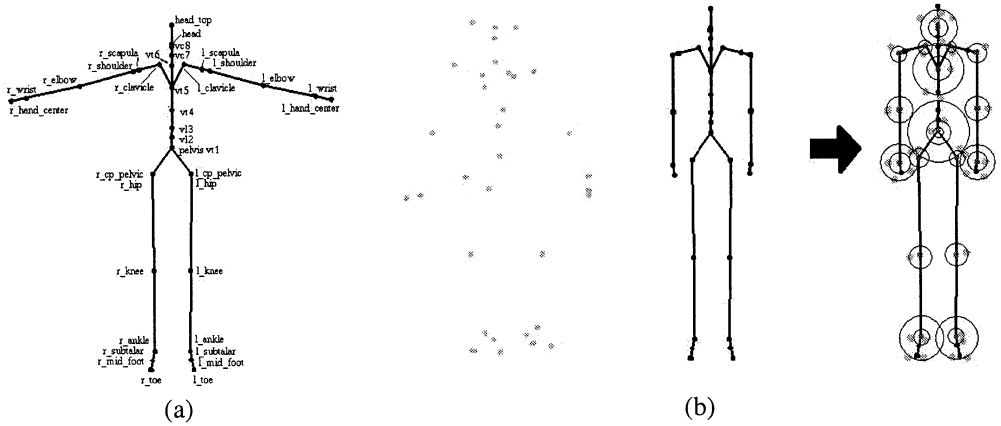


Figure 3: Skeleton and Marker Model. (a) Generic skeleton Model. (b) The generic model is scaled to conform the performer's anatomy. Each marker is attached to a joint and can move on a sphere centered around that joint.

these markers are reconstructed by trinocular stereo, that is, using at least three cameras. This is in contrast to markers reconstructed using only two camera views, and for which the projections into the other views failed.

Once we have reconstructed these trinocular 3–D markers in the first frame, we need to compare the number of reconstructed markers with the number of markers known to be carried by the actor. As all remaining processing is automatic, it is absolutely essential that all markers be identified in the first frame. Any marker not present in the first frame is lost for the entire sequence. Therefore, if the number of reconstructed markers is insufficient, a second stereo matching is performed, this time also taking into account markers seen in only two views. As binocular stereo matching is bound to introduce errors, the user is then prompted to confirm whether or not these binocular reconstructions are correct.

As soon as all markers are found in the first frame, the user is asked to associate each marker to a joint. For each highlighted marker, the user must select a body part and corresponding joint. Any marker not associated to a body part is discarded during the fitting process. Once these associations have been manually created, we can proceed with 2–D and 3–D tracking of the markers over the entire sequence. 2–D tracking is carried out at the same time as 3–D tracking because 2–D sequences are bound to provide more continuity than reconstructed 3–D sequences. We therefore use 2–D tracking in order to accelerate 3–D reconstruction: For each reliably reconstructed marker in frame [f], we consider the two sets of 2–D coordinates that were used to compute its 3–D coordinates. After 2–D tracking, these two sets of 2–D coordinates will most likely have links to two sets of 2–D coordinates in [f+1], the next frame. If so, we can then use them in [f+1] to construct the corresponding 3–D marker. To determine the related 2–D positions in the other camera views, we reproject the 3–D coordinates, as in the stereo matching process described above. 3–D tracking propagates the information attached to each marker in the first frame throughout the entire gym motion, so that as many markers as possible are identified in all frames. A broken link in the tracked trajectory of a marker implies the loss of its identity and the user must then be prompted. In Section 3.2, we will see how we use the skeleton to overcome that problem in an automated fashion.

To compute the trajectory of a marker from frame [f] into frame [f+1], both in 2–D and 3–D, we look at the displacement of the marker over a four-frame sliding window (Malik et al., 1993). The basic assumption is that displacement is minimal from one frame into the next, and the idea is to predict and confirm the position of a marker in the next frame. The displacement of a marker from [f-1] into [f] predicts the position in [f+1]. The actual position in [f+1] and the projection of the movement into [f+2] should confirm the previously-made hypothesis by eliminating ambiguities.

At the end of the marker reconstruction process and 2–D/3–D tracking steps, we have the gym motion reconstructed in 3–D, the trajectories of the markers throughout the sequence, as well as the identification of the markers with respect to the skeleton model.

3.1.2 Initial Joint Localization Let us consider a referential bound to a bone represented as a segment. Under the assumption that the distance between markers and joints remains constant, the markers that are attached on adjacent segments move on a sphere centered on the joint that links the two segments. The position of a segment in space is completely defined by three points. Thus, if we have a minimum of three markers on a segment, we can define the position and orientation of that segment in space. Afterwards, we compute the movement of the markers on adjacent segments in the referential established by these markers and we estimate their centers of rotation (Silaghi et al., 1998).

To take advantage of this observation, we partition the markers into sets that appear to move rigidly and estimate the 3–D location of the center of rotation between adjacent subsets, which corresponds to the joint location. This yields the

approximate 3–D location of thirteen major joints, namely the joints of the arms and legs, as well as the location of the pelvic joint, at the base of the spine.

3.1.3 Skeleton Initialization Given these thirteen joint locations in all frames, we take the median distances between them to be estimates of the length of the performer's limbs. We then use anthropometric tables to infer the length of the other skeleton segments. This gives us a skeleton model scaled to the size of the actor. This model, however, is a static one. It has the appropriate dimensions but does not yet capture the postures for the gym sequence or the relative position of markers and joints.

To estimate those distances, we first need to roughly position the skeleton in each frame by minimizing the distance of the thirteen key joints to the corresponding centers of rotation. This is done by minimizing an objective function that is the sum of square distances from the centers of rotation to the joint it is attached to. Given the fact that we use a sampling rate of 100 Hertz and that the gym motion is slow, the displacement from one frame to another is very small. Fitting is performed one frame at a time, and the initial parameter values for frame [f] are the optimized parameters obtained from the fitting in the previous frame [f-1]. As we only have thirteen observations for each frame, we do not attempt to estimate all of the skeleton's degrees of freedom. Only ten joints (shoulders, elbows, hips, knees, pelvic joint and the fourth spine vertebra) are active while all the others remain frozen. This yields the postures of the skeleton in all frames of the gym motion. In other words, we now have values of the global positioning vectors and degrees of freedom in each frame, as well as a better approximation to the limb lengths of the skeleton.

3.1.4 Global Fitting We now have a skeleton model that is scaled to the size of the performing actor, but we are still missing a complete marker model, that is one that specifies where the markers are positioned on the actor's body and their distance to the joints to which they are attached. This is computed by performing a second least-squares minimization where the actual 3–D marker locations become the data to which we intend to fit the skeleton.

Markers are not located exactly on the joints and the marker-to-joint distances must be estimated. To this end, we superimpose the markers' 3–D coordinates with the previously computed skeleton postures. In each frame, we then compute the distance from the marker to the joint and we take the median value of these distances to be our initial estimate of the marker-to-joint distance. Taking the marker model to be the distance from marker to joint means that the marker is expected to always be located on a sphere centered at the joint. We now have all the information required to fit the skeleton model to the observation data. The initial state is given by the previously obtained skeleton postures. As we need to check that all markers are present and identified before fitting, we do it one frame at a time.

For each frame and for each marker, once the fitting is complete, the distance between marker and joint is stored. At the end of the gym motion sequence, we have as many such distances per marker as there are frames. The median value of these distances is an improved approximation of the marker-to-joint distance and becomes the final marker model.

3.2 Capturing Complex Motions

The resulting skeleton-and-marker model can now be applied to motions that we actually wish to capture. The procedure is very similar to the one used in the global fitting step of the previous section. However, we are now dealing with potentially complex motions. Consequently, even though 2–D and 3–D tracking ensure the identification of a large number of markers from one frame to another, ambiguities, sudden acceleration or occlusions will often cause breaks in the tracking links or erroneous reconstructions. For this reason, it has proved to be necessary to increase our procedure's robustness by using the skeleton to drive the reconstruction process, as discussed below.

The user is once again required to identify the markers in the first frame. However, he will no longer be associating 3–D markers to joints, but directly to 3–D markers located on the body model as computed during the calibration phase.

3.2.1 Skeleton Based Tracking In order to improve the results of stereo matching, we use the skeleton for applying a visibility and occlusion test to each pair of 2–D markers used to construct a 3–D marker, thus verifying the validity of the reconstruction.

Visibility Check A marker is expected to be visible in a given view if it is seen more or less face on as opposed to edge on, that is if the surface normal at the marker's location and the line of sight form an acute angle. Suppose that we have reconstructed a certain 3–D marker using the 2–D pair (marker i1, view j1) and (marker i2, view j2); we check that these two markers i1 and i2 are indeed visible in views j1 and j2 respectively. Still assuming that displacement is minimal from one frame to the next, we use the skeleton's posture in the previous frame and calculate the normal at the 3–D marker's location with respect to its underlying body part segment. We draw the line joining the 3–D marker coordinates to the position in space of the camera and if the angle between the normal and the line is acute, then the marker is visible. If this test shows that we have used the wrong 2–D coordinates for reconstruction, we must select other candidate 2–D coordinates: As discussed in Section 3.1.1, each 3–D marker is associated to two sets of 2–D coordinates

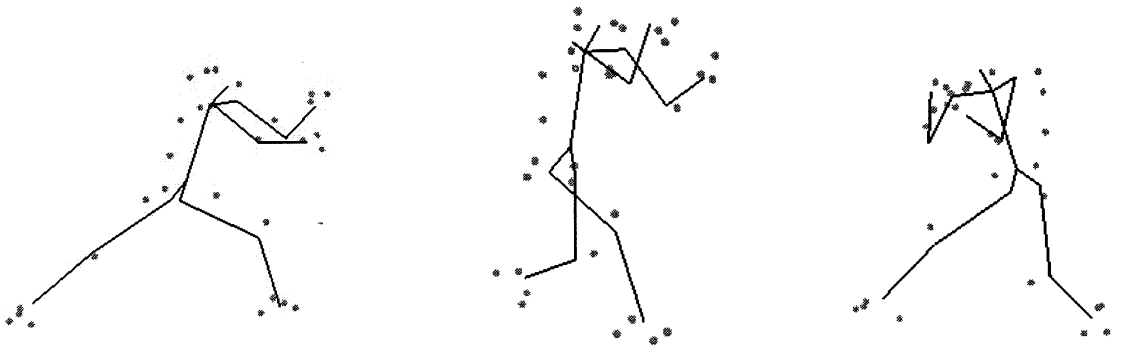


Figure 4: Three different frames from the karate motion (set 30 frames apart), seen from various viewpoints.

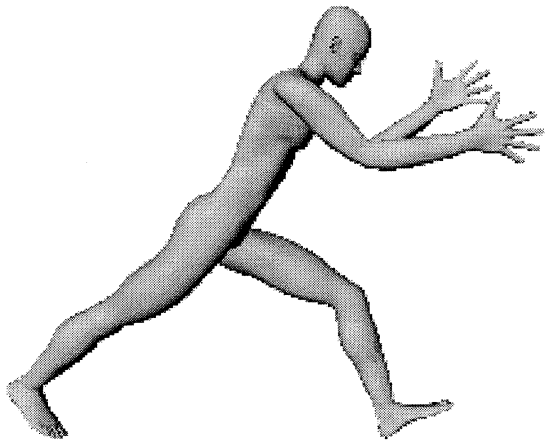


Figure 5: Virtual actor performing one of the recovered motions.

determined by stereo correspondence, which we then use for reconstructing the 3–D marker. To this 3–D marker, we then also associate the 2–D coordinates from the remaining camera views onto which the 3–D coordinates of the marker projected correctly. Given that the visibility test has detected an erroneous 3–D reconstruction, we choose one of the 2–D coordinates computed via 3–D to 2–D projection, and calculate new 3–D coordinates. We then perform a new visibility test, and if this fails, we repeat the entire procedure.

Occlusion check Once a 3–D marker has passed the visibility test, it needs to undergo the occlusion check: We want to ensure that the 3–D marker is not occluded from some camera views by another body part. To this end, we approximate body parts by solids, cylinders for limbs and a sphere for the head. In the case of limbs, the cylinder's axis is the corresponding bone and the radius is the average joint-to-marker distance of the markers associated to this body part. In the case of the sphere, the center is the mid-point of the segment. For each 3–D marker, a line is traced from the marker to the position of the camera, and tested for intersection with all body part solids. In case of intersection with a solid, the marker is most likely occluded from this camera view. Therefore, we conclude that we have used erroneous 2–D coordinates for reconstruction. As before, we choose other 2–D coordinates and repeat the process.

3.2.2 Marker inventory When all the markers have been reconstructed and tested, we can proceed with tracking and fitting. More specifically, for each frame, we perform 3–D reconstruction, tracking from the previous frame into the present one, identification of all markers, and finally, fitting of the skeleton-and-marker model to the observations. In order for the fitting to work correctly, all markers must be present in every frame. To ensure this, we carry out a marker inventory after 3–D reconstruction and before fitting. Say we have just performed 3–D reconstruction using the 2–D data of frame [f], and we have thus obtained a set of markers. We then proceed with the following checks:

1. If the number of markers reconstructed using trinocular stereo is smaller than the actual number of markers worn by the actor, we perform binocular reconstruction and add the newly calculated coordinates to the already existing list of markers.

2. We perform 3–D tracking from [f-1] into [f], thus identifying a certain number of markers in [f], i.e. attaching them to their legitimate joint.

3. If all markers are still not found, we attempt to identify the 3–D markers that are still anonymous. We find all the skeleton's joints that are missing one or more markers. Assuming that displacement is minimal from one frame to another, we retrieve the coordinates of these joints in the previous frame, and calculate the distance from these joints to each remaining unidentified 3–D marker; the distance closest to the marker-to-joint distance specified by the marker model yields an association of the 3–D marker to that joint.

4. If the distance from marker to joint is larger than the distance specified by the marker model, we "bind" the coordinates of the 3–D marker to the joint: We change its 3–D coordinates so that the marker moves within an acceptable distance of the joint. We however leave all reliably reconstructed 3–D markers untouched.

5. In the worst-case scenario, there may still be joints that are missing markers. We retrieve these markers in the three previous frame [f-3], [f-2] and [f-1], and calculate the acceleration; we apply this acceleration to the position in [f-1], thus obtaining an estimated position of the marker in the current frame [f]. As before, we calculate the distance from this inferred position to its associated joint. If it is out of range, we "bind" the coordinates.

3.3 Fitting Results

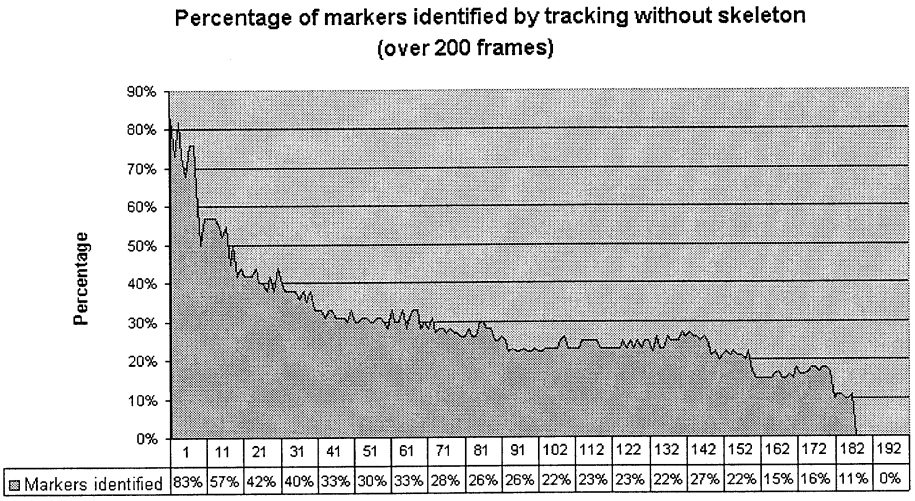


Figure 6: Percentage of markers identified by simple tracking, for the karate motion of Figurefig:SkeletonResults. Without skeleton-based tracking, most of them are quickly lost.

Figures 4 and 5 show the results obtained for a difficult karate motion that involves complex movements and sudden accelerations. Using the skeleton has enabled us to improve every step of the process, from 3–D reconstruction, to tracking and identification of the markers. It is robust with respect to noisy data: Out-of-bound and non-identified markers are rejected and occluded markers are properly handled. The effectiveness of skeleton-based tracking is illustrated by Figure 6: For the karate motion of Figure 4, our system does not loose any marker whereas, if we were to use only the simple tracking described above, we would quickly loose most of them.

4 VIDEO-BASED MODELING

As video cameras become increasingly prevalent, for example as attachment to most computers, video-based approaches become increasingly attractive means of deriving models of people such as clones for video-conferencing purposes. Such approaches also allow the exploitation of ordinary movies to reconstruct the faces and bodies of actors or famous people that cannot easily be scanned using active techniques, for example because they are unavailable or long dead. In the remainder of this section, we first discuss our approach to face and then body modeling.

4.1 Face Modeling

Given a set of potentially uncalibrated images or a video sequence, our goal is to fit the animation mask of Figure 1(a). Our challenge, here, is to solve the structure from motion problem in a case where

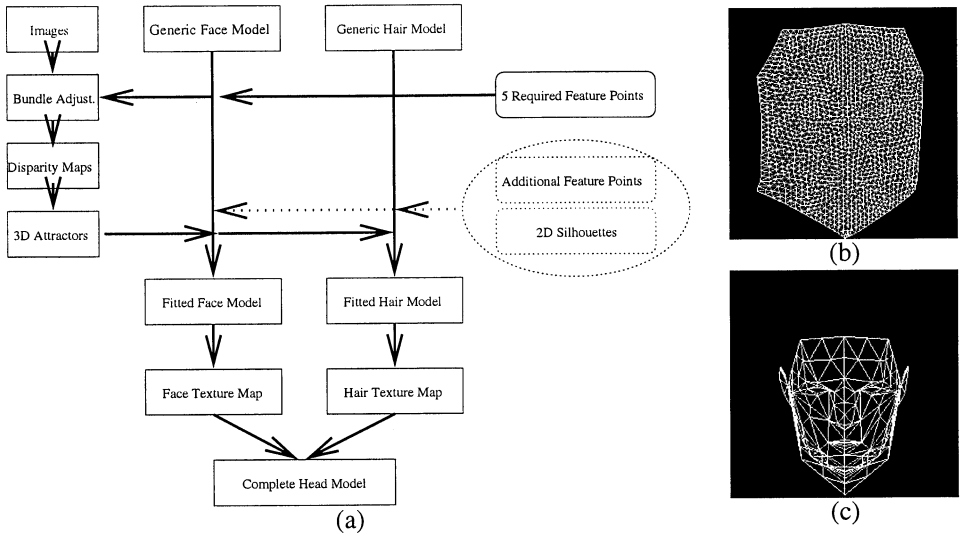


Figure 7: Face Reconstruction Procedure: (a) Flow chart. The manually and semi-automatically entered data appears on the right. The location of 5 2-D points must be supplied, the rest is optional. (b) Regular sampling of the face used to perform bundle adjustment. (c) Control triangulation used to deform the face.

- Correspondences are hard to establish and can be expected to be neither precise nor reliable due to lack of texture.
- A Euclidean or Quasi-Euclidean (Beardsley et al., 1997) reconstruction is required for realism.
- The motion is far from being optimal for most of the auto-calibration techniques that have been developed in recent years (Sturm, 1997, Zisserman et al., 1998).

To overcome these difficulties, we have developed an approach based on bundle-adjustment that takes advantage of our rough knowledge of the face's shape, in the form of the generic face model of Section 2 to introduce regularization constraints. This has allowed us to robustly estimate the relative head motion. The resulting image registration is accurate enough to use a simple correlation-based stereo algorithm to derive 3–D information from the data and to fit the animation model to it.

Bundle-adjustment is, of course, well established technique in the photogrammetric community (Gruen and Beyer, 1992). However, it is typically used in a context, mapping or close-range photogrammetry, where reliable and precise correspondences can be established. In addition, because it involves nonlinear optimization, it requires good initialization for proper convergence. Lately, it has been increasingly used in the computer vision community to refine the output of auto-calibration techniques. There again, however, most results have been demonstrated in man-made environments where feature points can be reliably extracted and matched across images. One cannot assume that those results carry over directly in the case of ill-textured objects such as faces and low quality correspondences.

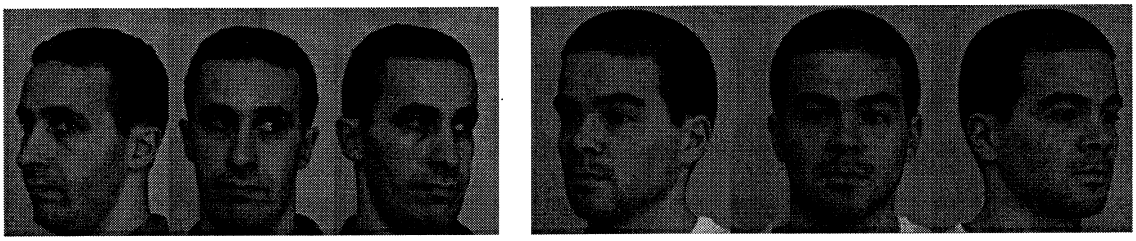
Successful approaches to automating the fitting process have involved the use of optical flow (DeCarlo and Metaxas, 1998) or appearance based techniques (Kang, 1997) to overcome the fact that faces have little texture and that, as a result, automatically and reliably establishing correspondences is difficult. This latter technique is closely related to ours because head shape and camera motion are recovered simultaneously. However, the optical flow approach avoids the "correspondence problem" at the cost of making assumptions about constant illumination of the face that may be violated as the head moves. This tends to limit the range of images that can be used, especially if the lighting is not diffuse. More recently, another extremely impressive appearance-based approach that uses a sophisticated statistical head model has been proposed (Blanz and Vetter, 1999). This model has been learned from a large database of human heads and its parameters can be adjusted so that it can synthesize images that closely resemble the input image or images. While the result are outstanding even when only one image is used, the recovered shape cannot be guaranteed to be correct unless more than one is used. Because the model is Euclidean, initial camera parameters must be supplied when dealing with uncalibrated imagery. Therefore, the technique proposed here could be used to initialize the Blanz & Vetter system in an automated fashion. In other words, if we had had their model, we could have used it to develop the technique described here.

Our procedure takes the steps depicted by the flowchart of Figure 7 and described in more detail below. The only manual intervention that is mandatory is supplying the approximate 2–D location on one single image of five feature points: Corners of the eyes and mouth and tip of the nose.

4.1.1 Relative Motion Recovery First, we estimate the relative motion of the face with respect to the camera. Given sequences in which the subjects keep a fairly neutral facial expression, we treat the head as a rigid object. We assume that the intrinsic camera parameters remain constant throughout the sequence. In theory, given high precision matches, bundle-adjustment can recover both intrinsic parameters and camera motion (Gruen and Beyer, 1992). The same holds true for recent auto-calibration techniques but, typical sequences of head images are close to exhibiting degenerate motions (Sturm, 1997, Zisserman et al., 1998). Again, extremely precise matches would be required.

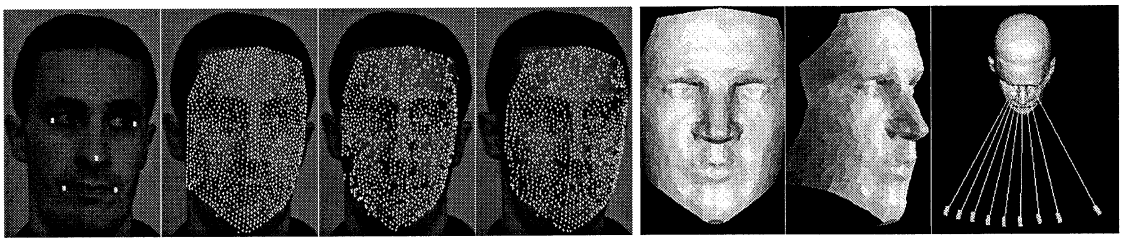
In practice, however, face images exhibit little texture and we must be prepared to deal with the potentially poor quality of the point matches. Therefore, we have chosen to roughly estimate the intrinsic parameters and to concentrate on computing the extrinsic ones using bundle-adjustment: We use an approximate value for the focal length and assume that the principal point remains in the center of the image. By so doing, we generate 3–D models that are deformed versions of the real heads. When the motion between the camera viewpoints is a pure translation, this deformation is an affine transform (Luong and Viéville, 1996). In practice, the deformation is still adequately modeled by an affine transform even if the motion is not a pure translation (Fua, 2000). The closer the approximate value of the focal length to its true value, the closer that affine transform is to being a simple rotation, translation and scaling.

Initialization A well known limitation of bundle adjustment algorithms is the fact that, to ensure convergence, one must provide initial adequate initialization. To fulfil this requirement, we begin by retriangulating the surface of the generic face model introduced in Section 2 to produce the regular mesh shown in Figure 7(b). We will refer to it as the *bundle-adjustment triangulation.*



(a) (b)

Figure 8: Input video sequence: For each person, 3 of a sequence of 9 consecutive images.



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

Figure 9: Regularized bundle-adjustment: (a) The five manually supplied keypoints used to compute the orientation of the first camera. (b) The projections of the bundle-adjustment triangulation vertices of Figure 7(b) into the central image of Figure 8(a). (c,d) Matching point in the images immediately following and immediately preceding the central image of Figure 8(a) in the sequence. (e,f) Shaded representation of the bundle-adjustment triangulation. (g) Recovered relative camera positions.

To initialize the process for a video sequence such as the ones shown in Figure 8, we manually supply the approximate position of the five feature points depicted by Figure 9(a) in one, and only one, reference image. We compute the position and orientation of the reference camera that brings the five projections of the corresponding keypoints as close as possible to those positions. We then estimate the positions and orientations for the two images on either side of the reference image as follows.

Generic Bundle Adjustment As shown in Figure 9(b), our initial orientation guarantees that the bundle-adjustment triangulation vertices' projections fall roughly on the face. We match these projections into the other images using a simple correlation-based algorithm. Figure 9(c,d) depicts the results. For each of vertex (x_i, y_i, z_i) of the bundle-adjustment triangulation and each projection (u_i^j, v_i^j) of this vertex in image j, we write two *observation equations*:

$$\begin{aligned} Pr_u^j(x_i + dx_i, y_i + dy_i, z_i + dz_i) &= u_i^j + \epsilon_{u_i^j} \\ Pr_v^j(x_i + dx_i, y_i + dy_i, z_i + dz_i) &= v_i^j + \epsilon_{v_i^j} \end{aligned} \quad (1)$$

where $Pr_u^j(x,y,z)$ and $Pr_v^j(x,y,z)$ denote the two image coordinates of the projection of point (x,y,z) in image j using the current estimate of the camera models; (dx_i, dy_i, dz_i) represents the 3–D displacement of vertex i to conform to the actual face shape; and, $\epsilon_{u_i^j}, \epsilon_{v_i^j}$ are the projection errors to be minimized. The camera position parameters can be recovered by minimizing the sum of the squares of the $\epsilon_{u_i^j}$ and $\epsilon_{v_i^j}$. with respect to the six external parameters of each camera and the (dx_i, dy_i, dz_i) displacement vectors. The solution can only be found up to a global rotation, translation and scaling. To remove this ambiguity, we fix the position of the first camera and one additional parameter such as the distance of one vertex in the triangulation.

Robust Bundle Adjustment If the correspondences were perfect, the above procedure would suffice. However, the point correspondences can be expected to be noisy and to include mismatches. To increase the procedure's robustness, we introduce the two following techniques.

Iterative reweighted least squares. We first run the bundle adjustment algorithm with all the observations of Equation 1 having the same weight. We then recompute these weights so that they are inversely proportional to the final residual errors. We minimize our criterion again using these new weights and iterate the whole process until the weights stabilize.

Regularization. We prevent excessive deformation of the bundle-adjustment triangulation by treating the bundle-adjustment triangulation's facets as C^0 finite elements and adding a quadratic regularization term to the sum of the squares of the $\epsilon_{u_i^j}$ and $\epsilon_{v_i^j}$ of Equation 1.

For the image triplet formed by the central image of the video sequence of Figure 8(a) and the images immediately preceding and following it, the procedure yields the bundle-adjustment triangulation's shape depicted by Figure 9(e,f). By repeating this computation over all overlapping triplets of images in the video sequences we can compute the camera positions depicted by Figure 9(g).

4.1.2 Model Fitting Given the camera models computed above, we can now recover additional information about the surface by using a simple correlation-based algorithm (Fua, 1993) to compute a disparity map for each pair of consecutive images in the video sequences and by turning each valid disparity value into a 3–D point. Because, these 3–D points typically form an extremely noisy and irregular sampling of the underlying global 3–D surface, we begin by robustly fitting surface patches to the raw 3–D points. This first step eliminates some of the outliers and generates meaningful local surface information for arbitrary surface orientation and topology (Fua, 1997).

Our goal, then, is to deform the generic mask so that it conforms to the cloud of points, that is to treat each patch as an attractor and to minimize its distance to the final mask. In our implementation, this is achieved by computing the orthogonal distance d_i^a of each attractor to the closest facet as a function of the x,y, and z coordinates of its vertices and minimizing the objective function:

$$\mathcal{E} = \sum_i (d_i^a)^2 \quad . \tag{2}$$

Control Triangulation In theory we could optimize with respect to the state vector P of all x, y, and z coordinates of the surface triangulation. However, because the image data is very noisy, we would have to impose a very strong regularization constraint. Instead, we introduce *control triangulations* such as the one shown in Figure 7(c). The vertices of the surface triangulation are "attached" to the control triangulation and the range of allowable deformations of the surface triangulation is defined in terms of weighted averages of displacements of the vertices of the control triangulation (Fua and Miccio, 1998).

Because there may be gaps in the image data, it is necessary to add a small stiffness term into the optimization to ensure that the displacements of the control vertices are consistent with their neighbors where there is little or no data. As before, we treat the control triangulation's facets as C^0 finite elements and add a quadratic stiffness term the objective function of Equation 2.

Because there is no guarantee that the image data covers equally both sides of the head, we also add a small number of symmetry observations between control vertices on both sides of the face. They serve the same purpose as the stiffness term: Where there is no data, the shape is derived by symmetry. An alternative would have been to use a completely symmetric model with half of the degrees of freedom of the one we use. We chose not to do so because, in reality, faces are somewhat asymmetric. Because the control triangulation has fewer vertices that are more regularly spaced than the surface triangulation, the least-squares optimization has better convergence properties. Of course, the finer the control triangulation, the less smoothing it provides. By using a precomputed set of increasingly refined control triangulations, we implement a hierarchical fitting scheme that has proved very useful when dealing with noisy data We recompute the facet closest to each attractor at each stage of our hierarchical fitting scheme, that is each time we introduce a new control triangulation. To discount outliers, we also recompute the weight associated with each attractor and take it to be inversely proportional to the initial distance of the data point to the surface triangulation.

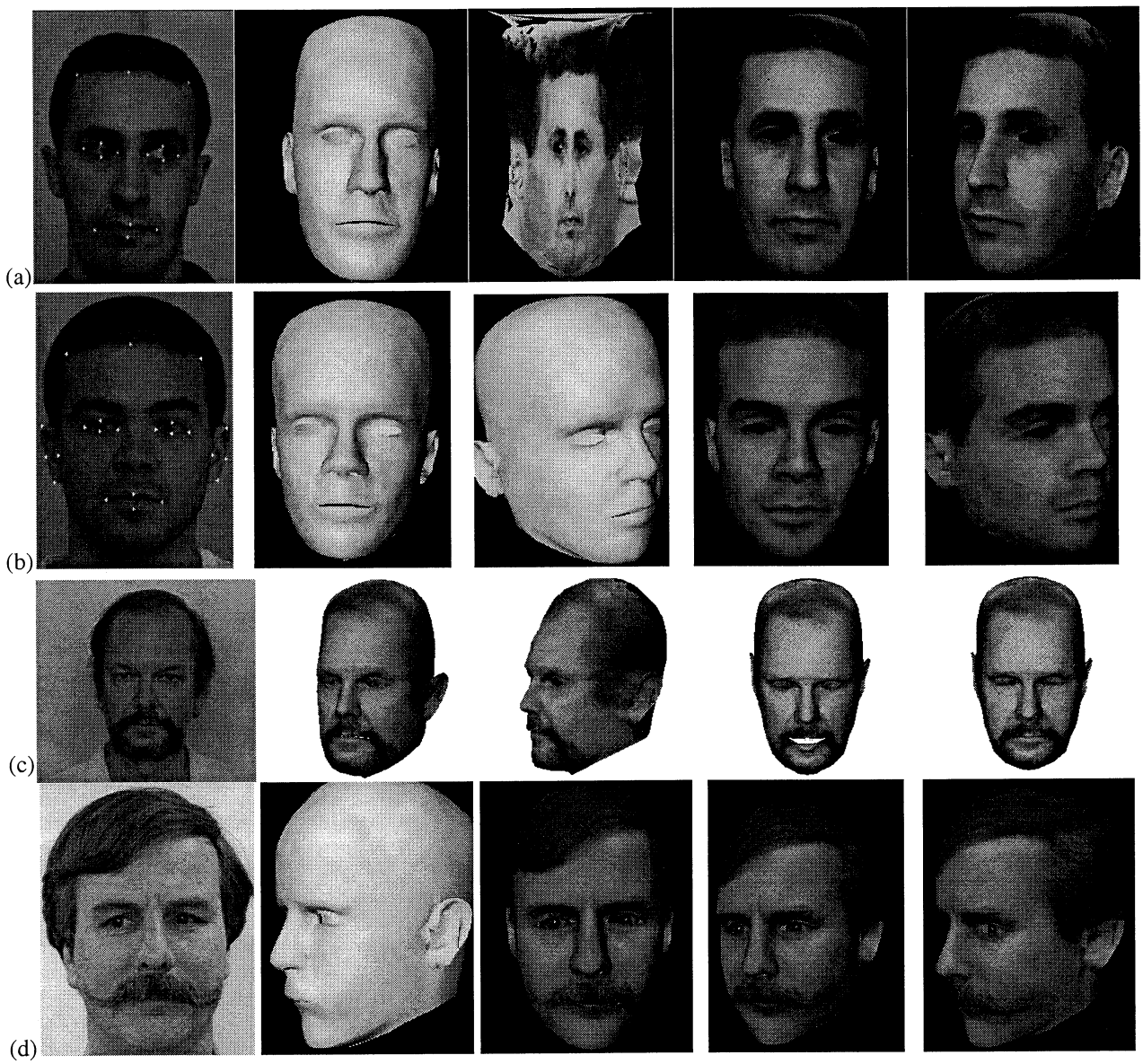


Figure 10: Reconstruction and animation. (a) For the subject of Figure 8(a): The manually supplied 2–D features points; two shaded views of the complete head model; the cylindrical texture map; and, textured model. (b) For the subject of Figure 8(b), the manually supplied 2–D features points; two shaded and two textured views of the head model. (c) Man with a beard: two texture-mapped views of the corresponding head model; and two synthetic expressions, opening of the mouth and closing of the eyes. (d) Man with a mustache: One shaded and three texture-mapped views of the corresponding head.

Shape and Texture Figure 10 depicts the output of our modeling procedure. In all cases, the head models can be animated. To ensure that some of the key elements of the face—corners of the eyes, mouth and hairline—project at the right places, we have manually supplied the location of the projection in one image of a few feature points such as the ones shown in the first column of Figure 10: We add to the objective function of Equation 2 a term that forces the projection of the generic mask's corresponding vertices to be close to them (Fua and Miccio, 1998). Note that the five manually supplied points used to initialize the bundle-adjustment procedure of Section 4.1.1 form a subset of these feature points. To produce these face models, the manual intervention required therefore reduces to supplying these few points by clicking on their approximate locations in one single image, which can be done quickly.

Because stereo tends to fail in the hair regions, the shape of the top of the head has been recovered by semi-automatically delineating in each image of the video sequence the boundary between the hair and the background and treating it as a silhouette that constrains the shape (Fua and Miccio, 1998). Given the final head model, the algorithm creates a cylindrical texture map, such as the one shown in the top row of Figure 10.

For a quantitative evaluation of these results, we have acquired 3–D models of the faces of both subjects of Figure 8 using a Minolta^{tm} scanner. The theoretical precision of the laser is approximately 0.3 millimeters and it can therefore be considered as a reasonable approximation of ground truth. In practice, even the laser exhibits a few artifacts. However,

since the two reconstruction methods are independent, places of agreement are very likely to be correct for both. As discussed in Section 4.1.1, we model the deformation induced by our arbitrary choice of internal camera parameters as an affine transform: We have therefore computed the affine transform that brings the bundle-adjustment triangulation closest to the laser-scanner model. In both cases, the median distance between the affine-transformed face models and the laser output is approximately 1 millimeter which, given the camera geometry, corresponds to a shift in disparity of less than 1/5 a pixel. The precision of the correlation based algorithm we use is in the order of half a pixel, outliers excluded (Fua, 1993). We therefore conclude that our motion recovery algorithm performs an effective and robust averaging of the input data.

4.2 Body Modeling

If the face can be assumed to be relatively rigid as a long as the subject does not change his expression, when dealing with the body, one must take into account its articulated nature. Here, we use two or three video sequences acquired using synchronized cameras. The body model and the image data are used throughout the fitting process.

Recently, a number of techniques have been proposed (Kakadiaris and Metaxas, 1996, Gavrila and Davis, 1996, Lerasle et al., 1996, Bregler and Malik, 1998) to track human motions from video sequences. They are fairly effective but use very simplified models of the human body, such as ellipsoids or cylinders, that do not precisely model the human shape and would not be sufficient for a truly realistic simulation. By contrast, we use the full body model of Figure 1.

The algorithm goes through four steps that we summarize below. For a more complete description, we refer the interested reader to our earlier publications (D'Apuzzo et al., 1999, Plänkers et al., 1999).

Data Acquisition Clouds of 3–D points are derived from the input images using correlation-based stereo (Fua, 1993). Alternatively, we can use least-squares matching to derive these clouds (D'Apuzzo et al., 2000). Silhouette edges may be delineated in several key-frames or automatically generated for the whole sequence.

Initialization: We first initialize the model interactively in one frame of the sequence. The user has to enter the approximate position of some key joints, like shoulders, elbows, hands, hips, knees and feet. Here, it was done by clicking on these features in two images and triangulating the corresponding points. This initialization gives us a rough shape, i.e. a scaling of the skeleton, and an approximate posture of the model.

Tracking: At a given time step the *tracking* process adjusts the model's joint angles by minimizing an objective function. This modified posture is saved for the current frame and serves as initialization for the next one. The computing power of today's PCs allows for interactivity. If, for some reason, the algorithm loses track the user simply pauses the program, adjusts the posture interactively and hands the control back to the algorithm for further processing.

Fitting: The results from the *tracking* step serve as initialization for a *fitting* step. Its goal is to refine the postures in all frames and to adjust the skeleton and/or metaball parameters to make the model correspond more closely to the person. The *fitting* optimizes over all frames simultaneously, by minimizing the same objective function as before. This allows us to find a single set of parameters that describe a model that is consistent with the images of the whole sequence. The results are further improved by introducing inter-frame constraints such as smoothness or limits on velocity/acceleration.

In practice, the model and the constraints it imposes are used to overcome the inherent noisiness of the data. We recover both motion and body shape from stereo video sequences. The corresponding parameters can be used to recreate realistic 3–D animations.

4.2.1 Least Squares Framework Our system must deal with heterogeneous sources of information—3–D data and 2–D outlines—whose contributions may not be commensurate. To this end, we have developed the following framework.

In standard least-squares fashion, we use the image data to write $nobs$ observation equations of the form

$$f_i(S) = obs_i - \epsilon_i \quad , 1 \leq i \leq nobs \quad , \tag{3}$$

where S is the state vector that defines the shape and position of the body model and ϵ_i is the deviation from the model. We will then minimize

$$v^T P v \Rightarrow Min \ , \tag{4}$$

where v is the vector of residuals and P is a weight matrix associated with the observations. P is usually introduced as diagonal.

Our system must be able to deal with observations coming from different sources that may not be commensurate with each other. Formally, we can rewrite the observation equations of Equation 3 as

$$f_i^{type}(S) = obs_i^{type} - \epsilon_i \quad , 1 \leq i \leq nobs \quad , \tag{5}$$

with weight p_i^{type}, where $type$ is one of the possible types of observations we use. In this paper, $type$ can be object space coordinates or silhouette rays. However, other information cues can easily be integrated. The individual weights of the different types of observations have to be homogenized before estimation according to:

$$\frac{p_i^k}{p_j^l} = \frac{\left(\sigma_j^l\right)^2}{\left(\sigma_i^k\right)^2} , \tag{6}$$

where σ_j^l, σ_i^k are the a priori standard deviations of the observations obs_i, obs_j of type k, l. Applying least-squares estimation implies the joint minimum

$$\sum_{type=1}^{nt} v^{type} P_{type} v^{type} \Rightarrow Min , \tag{7}$$

with nt the number of observation types, which then leads to the well-known normal equations which need to be solved using standard techniques.

In practice, however, it is very difficult to estimate the standard deviations of Eq. 6. We therefore use the following heuristics, which has proved to be very effective. To ensure that the minimization proceeds smoothly we multiply the weight p_i^{type} of the n_{type} individual observations of a given type by a global coefficient w_{type} computed as follows:

$$\begin{aligned} G_{type} &= \frac{\sqrt{\sum_{1 \leq i \leq nobs, j=type} p_i^{type} \|\nabla f_i^j(S)\|^2}}{n_{type}} \\ w_{type} &= \frac{\lambda_{type}}{G_{type}} \end{aligned} \tag{8}$$

where λ_{type} is a user supplied coefficient between 0 and 1 that dictates the relative importance of the various kinds of observations. This guarantees that, initially at least, the magnitudes of the gradient terms for the various types have the appropriate relative values.

Since our overall problem is non-linear, the results are obtained through an iteration process. We use a sparse-matrix implementation of the Levenberg-Marquardt algorithm (Press et al., 1986) that can handle the large number of parameters and observations we must deal with.

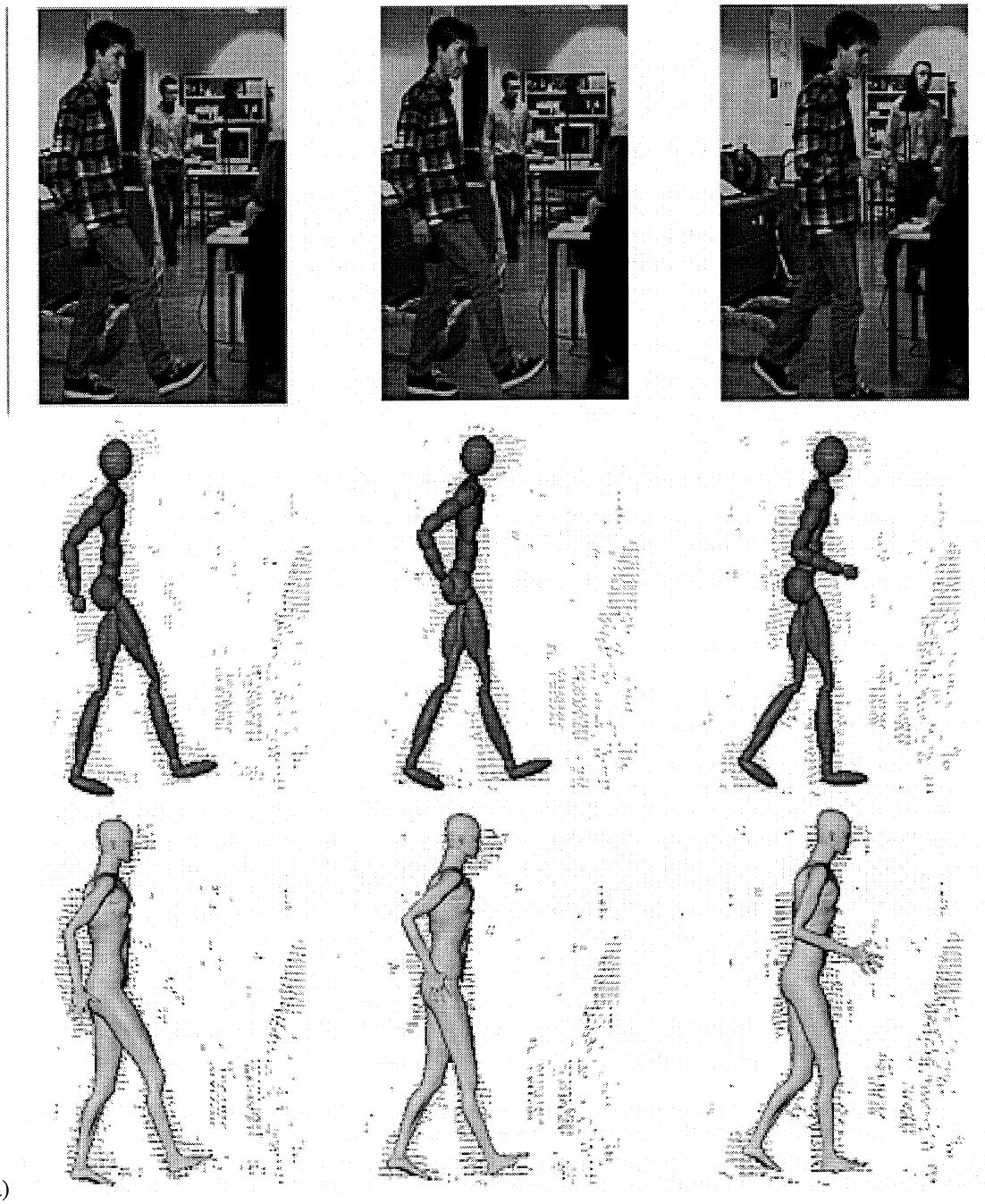
4.2.2 Skeleton Fitting and Motion Modeling The images of Figure 11 show somebody walking in front of a horizontally aligned stereo camera pair. The background and lighting were uncontrolled (standard office head lights) and the camera pair was about 5m from the person. The distance between the two cameras was 75cm. The images are interlaced and the processed half-frame has an effective resolution of 768×288. The disparities result in about 2000 3–D points, including reconstructed parts of the background. The top row of Figure 11 shows three frames out of 50 from this sequence. The result from the initial tracking process is depicted by the middle row of the Figure. The bottom row shows the output of the subsequent fitting step. Here, the dimensions of the skeleton and the size of the metaballs have been adjusted, resulting in slightly more realistic postures.

The sequence of Figure 12 exhibits complex motions of a naked upper body, taken with a camera set up in front of the subject. Three cameras in an L configuration took interlaced images at 20 frames/sec with an effective resolution of 432×288 per half-frame. Our stereo algorithm (Fua, 1993) produced very dense point clouds with about 4000 3–D points on the surface of the subject, even without textured clothes. To increase the frame rate and, thus, reduce the difference in posture between frames we used both halves of the interlaced images and adjusted the camera calibration accordingly.

Although this motion involves severe occlusions, the system faithfully tracks the arms and yields both body postures and an adapted skeleton. The technique of Section 4.1 was used to derive the model's head from one video-sequence.

5 CONCLUSION

We have presented a set of technique that allow us to fit complex facial and body animation models to potentially noisy data with minimal manual intervention. Consequently, using either optical motion capture data or video-sequences, these models can be instantiated robustly and quickly. Although the models were primarily designed for animation rather than fitting purposes, we have designed a framework that allows us to exploit them to resolve ambiguities in the image data.



(a)

Figure 11: Frames 0, 5 and 10 of a walking sequence. Top row: Original sequence. Middle row: Tracking results using the simple model, overlaid with the 3–D points' projections. Bottom Row: Final fitting results using the detailed model.

In future work, we intend to extend our approach to capturing facial dynamics in addition to body dynamics. We will use this motion capture data to characterize actions and derive animation and biometric models for specific motions, such as running, and specific facial expressions. Such a capability will improve our ability to visualize, analyze, edit and synthesize human motion. This will have applications ranging from movie making to sports medicine and athletic training.

ACKNOWLEDGEMENTS

We wish to thank Prof. Daniel Thalmann, Prof. Nadia Magnenat Thalmann, and Prof. Prem Kalra for having made their animation models available to us. We are also indebted to Prof. Grün for sharing with us his insights about least-squares technology.

The work reported here was funded in part by the Swiss National Science Foundation and in part under European Esprit project Motion Capture (MOCA).

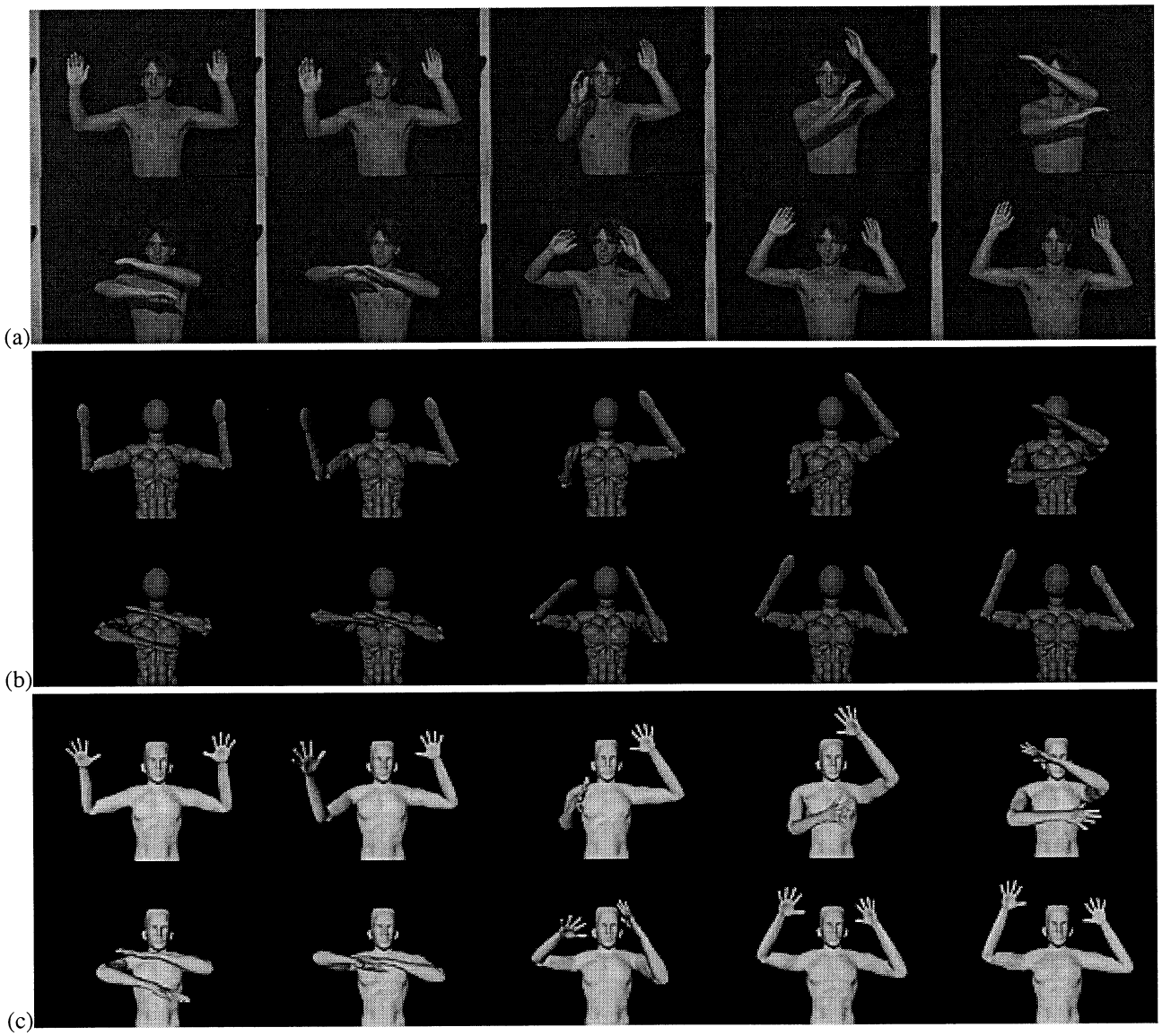


Figure 12: Upper body results. (a) Original sequence. (b, c) The final results after one tracking run to roughly adjust the posture and a fitting run to refine the skeleton and metaball proportions and the posture. The center row (b) depicts a discreet visualization of the metaballs and the bottom row (c) shows the smoothly rendered implicit surface.

REFERENCES

Beardsley, P. A., Zissermam, A. and Murray, D. W., 1997. Sequential update of projective and affine structure from motion. International Journal of Computer Vision 23(3), pp. 235–259.

Blanz, V. and Vetter, T., 1999. A Morphable Model for The Synthesis of 3–D Faces. In: Computer Graphics, SIGGRAPH Proceedings, Los Angeles, CA, pp. 71–78.

Bregler, C. and Malik, J., 1998. Tracking people with twists and exponential maps. Conference on Computer Vision and Pattern Recognition.

D'Apuzzo, N., Gruen, A., Plänkers, R. and Fua, P., 2000. Least Squares Matching Tracking Algorithm for Human Body Modeling. In: XIX ISPRS Congress, Amsterdam, Netherlands.

D'Apuzzo, N., Plänkers, R., Fua, P., Gruen, A. and Thalmann, D., 1999. Modeling Human Bodies from Video Sequences. In: Electronic Imaging, SPIE Photonics West Symposium, San Jose, CA.

DeCarlo, D. and Metaxas, D., 1998. Deformable Model-Based Shape and Motion Analysis from Images using Motion Residual Error. In: International Conference on Computer Vision, Bombay, India, pp. 113–119.

Faugeras, O. and Robert, L., 1996. What can two images tell us about a third one? International Journal of Computer Vision (18), pp. 5–19.

Ferrigno, G. and Pedotti, A., 1985. Elite: A digital dedicate hardware system for movement analysis via real-time tv signal processing. IEEE Transactions on Biomedical Engineering.

Fua, P., 1993. A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features. Machine Vision and Applications 6(1), pp. 35–49.

Fua, P., 1997. From Multiple Stereo Views to Multiple 3–D Surfaces. International Journal of Computer Vision 24(1), pp. 19–35.

Fua, P., 2000. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. International Journal of Computer Vision. In Press.

Fua, P. and Miccio, C., 1998. From Regular Images to Animated Heads: A Least Squares Approach. In: European Conference on Computer Vision, Freiburg, Germany, pp. 188–202.

Gavrila and Davis, L., 1996. 3d model-based tracking of humans in action : A multi-view approach. In: Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 73–80.

Gruen, A. and Beyer, H., 1992. System Calibration through Self-Calibration. In: Calibration and Orientation of Cameras in Computer Vision, Washington D.C.

Herda, L., Fua, P., Plänkers, R., D., Boulic, R. and Thalmann, D., 2000. Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion. Technical report, DI-LIG, EPFL. http://ligwww.epfl.ch/ plaenker/herda-et-al-ca00.pdf.

Kakadiaris and Metaxas, 1996. Model based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In: Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 81–87.

Kalra, P., Mangili, A., Thalmann, N. M. and Thalmann, D., 1992. Simulation of Facial Muscle Actions Based on Rational Free Form Deformations. In: Eurographics.

Kang, S. B., 1997. A Structure from Motion Approach using Constrained Deformable Models and Apperance Prediction. Technical Report CRL 97/6, Digital, Cambridge Research Laboratory.

Lerasle, F., Rives, G., Dhome, M. and Yassine, A., 1996. Human Body Tracking by Monocular Vision. In: European Conference on Computer Vision, Cambridge, England, pp. 518–527.

Luong, Q.-T. and Viéville, T., 1996. Canonical Representations for the Geometries of Multiple Projective Views. Computer Vision and Image Understanding 64(2), pp. 193–229.

Malik, N., Dracos, T. and Papantoniou, D., 1993. Particle tracking in three-dimensional turbulent flows - Part II: Particle tracking. Experiments in Fluids 15, pp. 279–294.

Plänkers, R., Fua, P. and D'Apuzzo, N., 1999. Automated Body Modeling from Video Sequences. In: ICCV Workshop on Modeling People, Corfu, Greece.

Press, W., Flannery, B., Teukolsky, S. and Vetterling, W., 1986. Numerical Recipes, the Art of Scientific Computing. Cambridge U. Press, Cambridge, MA.

Shen, J. and Thalmann, D., 1995. Interactive shape design using metaballs and splines. In: Implicit Surfaces, Grenoble, France.

Silaghi, M.-C., Plänkers, R., Boulic, R., Fua, P. and Thalmann, D., 1998. Local and global skeleton fitting techniques for optical motion capture. In: Workshop on Modelling and Motion Capture Techniques for Virtual Environments, Geneva, Switzerland.

Sturm, P., 1997. Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction. In: Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 1100–1105.

Thalmann, D., J.Shen and Chauvineau, E., 1996. Fast Realistic Human Body Deformations for Animation and VR Applications. In: Computer Graphics International, Pohang,Korea.

Titanic Special Reprint, 1997. Cinefex.

Zisserman, A., Liebowitz, D. and Armstrong, M., 1998. Resolving ambiguities in auto-calibration. Philosophical Transactions: Mathematical, Physical and Engineering Sciences (A) 356(1740), pp. 1193 – 1211.