HUMAN BODY MODELING AND MOTION ANALYSIS FROM VIDEO SEQUENCES

Pascal Fua[1], Armin Gruen[2], Ralf Plänkers[1], Nicola D'Apuzzo[2] and Daniel Thalmann[1]

[1] Computer Graphics Lab (LIG), EPFL, Lausanne, Switzerland
[2] Institute of Geodesy and Photogrammetry, ETHZ, Zurich, Switzerland

Commission V, Special Interest Group on "Animation"

KEY WORDS: Human Animation, Image Sequences, 3-D Tracking, Stereo, Silhouettes, Model Fitting

ABSTRACT

We present a comprehensive concept to fit animation models to a variety of different data derived from multi-image video sequences. Our goal is to record dynamically the body surface of a human in motion and to model it for animation purposes. This includes setting up and calibrating a system of three CCD-cameras, extracting image silhouettes, tracking individual key body points in 3-D, and generating surface data by stereo or multi-image matching. All these observations are brought together under a joint least squares estimation system, from which the body model parameters are derived. This represents a first report concerning our concept. The presented data stems from individual tests and is highly incomplete. However, these results support strongly the chosen concept and will lead to further developments and refinements.

1. INTRODUCTION

Synthetic modeling of human bodies and the simulation of motion is a longstanding problem in animation and much work is involved before a near-realistic performance can be achieved. At present, it takes an experienced designer a very long time to build a complete and realistic model that closely resembles a specific person. Digital photogrammetry offers a means to obtain the necessary data faster and in a more realistic fashion. Our ultimate goal is to automate the process: Eventually the whole task should be performed quickly by an operator who is not necessarily an experienced graphics designer. We should be able to invite a visitor to our laboratory, make him walk in front of a set of cameras, and produce, within a single day, a realistic animation of himself.

We concentrate on a video-based approach because of its comparatively low cost and better control of the dynamic nature of the process. While laser scanning technology provides a fairly good surface description of a static object from a given viewpoint, videogrammetry allows us in addition to measure and track particular points of interest, such as joints, and to record and track surface and point features on and around the object. Dynamic tracking can also be achieved using systems based on active infra red markers or magnetic sensors. But the first are expensive and also involve image processing techniques while the others entail the use of cumbersome wiring and associated inaccuracies.

The problem to be solved is twofold: First, robustly extract image information from the data; second fit the animation models to the extracted information. In this paper, we use video sequences acquired with two or more synchronized CCD-cameras to extract:

- Trajectories of body movement: Individual prominent body points are tracked in 3-D throughout the sequence.
- Corresponding image patches: Wherever a body part faces two or more of the cameras, its shape can be effectively derived from stereo and multi-image techniques.
- Outlines: Wherever a body part slants away from the camera, a silhouette edge appears in the images and can be used to derive 3-D information about the surface.

The last two sources of information are therefore complementary: The former is unreliable where the surface slants away from the camera, which is precisely where silhouettes can be found.

However, these information sources are noisy and may include artifacts. We aim at using the animation models not only to represent the data but also to guide the feature extraction process which allows for a substantial gain in performance. This paper reports on some preliminary results for our project, including an approach for tracking marked points and techniques for extracting stereo and silhouette data. Furthermore, we describe the animation models we use and show that we can recover joint locations and rough shapes of the limbs from motion sequences. In future work, we will integrate all data sources and use this knowledge to initialize the complete model and optimize its shape.

2. TRACKING OF PROMINENT POINTS

Our approach to tracking is based on multi-image recording. For this early study, we have chosen to analyse the

movement of retroreflective points stuck on the skin. These points can be treated as single particles, so that the particle tracking velocimetry concept (Maas et al., 1992, Malik et al., 1992) can be used without any modifications.

2.1 Image acquisition

Three CCD cameras in a triangular arrangement (left, right, bottom) are used (Figure 1).
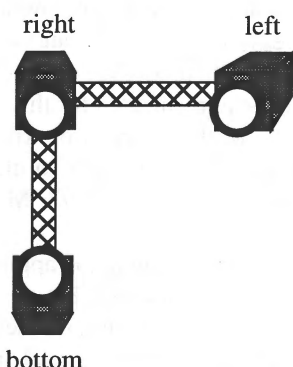

Fig. 1: Arrangement of the three CCD cameras

A sequence of triplet images is acquired with a frame grabber and the images are stored with 768x576 pixels at 8 bit quantisation. Figure 2 shows the images taken by the three cameras for the start and for the end frames of a sequence.
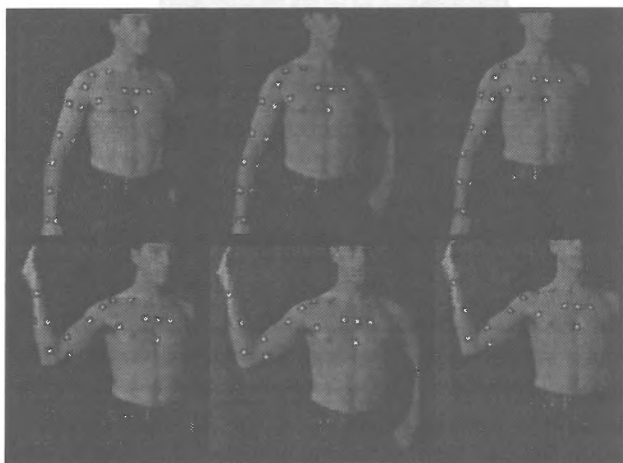

Fig. 2: Start (top) and end (bottom) frames of a sequence left: left view, centre: right view, right: bottom view

2.2 System calibration

A reference bar with two retroreflective target points is moved through the object space and at each location image triplets are acquired. The image coordinates of the two target points are measured with centroid operations for each triplet. The three camera system can then be calibrated by self-calibrating bundle adjustment with the additional information of the known distance between the two points at every location (Maas, 1998).

2.3 Determination of 3-D coordinates

The first task of the tracking process is the determination of the 3-D coordinates of the target points for each triplet of the sequence. In case of strong interlacing effects, the odd and the even lines of the images are treated separately. To detect the retroreflective points, the images are firstly filtered (high pass filter and then thresholding) and the coordinates of the candidate points in the images are determined by centroid operations. Once this process is done in the three images of all frames, the 3-D coordinates of the points can be computed by forward intersection. The result is a list of data sets Set_i $i=1...n$ (n number of triplets) which contain the 3-D point coordinates for each triplet.

2.4 Tracking process

The aim of the tracking process is now to derive the 3-D connections between the points through the sequence. The tracking system operates on three successive data sets Set_i, Set_{i+1} and Set_{i+2}. A point of the set Set_i firstly defines a three dimensional search volume for $i+1$ with the premise of a maximum velocity of the movement. When there are two or more candidates in the search volume, the feasible connections between Set_i and Set_{i+1} are extrapolated to Set_{i+2}, where a reduced search volume is defined with the premise of a maximum acceleration of the movement. If even in this reduced search volume two or more candidates are found, then the one with the smallest acceleration (i.e. the difference between the velocity vector in two adjacent frames) is preferred. This last rule is based on the observation that the trajectories are generally smooth.

Figure 3 shows part of the analysed sequence, taken by the right camera. The arm is firstly lifted, then twisted, then it is bended and at the end it returns a little bit.
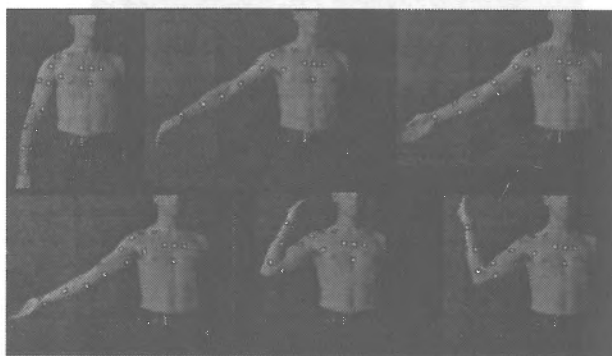

Fig. 3: Motion of the arm in a particalur frame (upper left to lower right)

The result of the tracking process forms a database of trajectories. Figure 4 shows the computed 3-D trajectories of the target points for the arm motion sequence.

Fig. 4: Computed 3-D trajectories of the target points

Obviously, the trajectories are broken. This is because points disappear and new points appear through the three distinguished steps of the arm motion (lift, twist, bend). The trajectories therefore do not continue for all points through the full sequence. Figure 5 shows the trajectories of the arm lift and arm bend motion projected back into one view. Tracking a complete and complex motion (e.g. connecting the lift of the arm with the twist and the bend) is at this stage of the project not yet reliably available, indeed it is an important task for the future work.
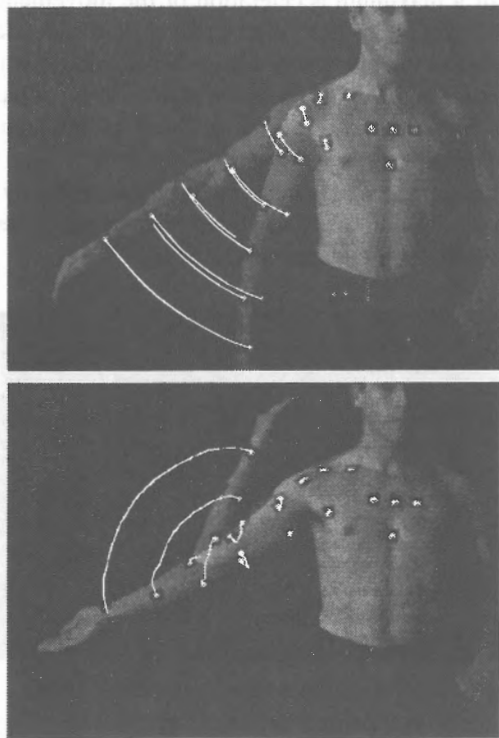

Fig. 5: Trajectories for arm lift and bend operations projected back into image frames

Also, we plan to solve this task without using retro-targets. In a further step we will have the person wear tight and well-textured textiles, which will allow us to track the textile pattern.

3. GENERATING SILHOUETTE DATA

This can be achieved in many ways. Many authors cite the use of Canny edge detectors in the images with subtracted background. This is an automatic but low-level method and thus relatively easy to implement but not very robust in practice. Automated silhouette edge detectors have been developed and could be implemented for this use (Vaillant and Faugeras, 1992). In this work, we experiment with semi-automated tools to allow the user to quickly sketch the silhouette edges (Grün and Li, 1997, Mortensen and Barett, 1995).

We have made first tests concerning the application of energy-minimizing functions (Snakes). Figure 6 shows the results of applying LSB-Snakes to the silhouette of an arm. Since, in general, silhouettes from several instantaneous frames do not form a unique space curve, we use the LSB-Snakes in their image space version.
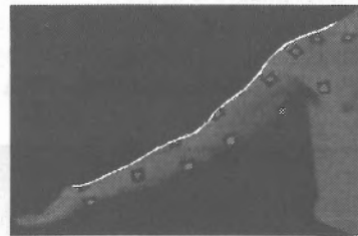

Fig. 6: Silhouette extraction with LSB-snakes

4. MODELS

In this section, we first describe the complete model that we use for animation purposes. This model has too many degrees of freedom to be effectively fit to noisy data without a-priori knowledge. We therefore introduce a simplified model that we have used to derive an initial shape and position. In future work, we will use this knowledge to initialize the complete one before refining it.

4.1 Complete Animation Model

Generally, virtual human bodies are structured as articulated bodies defined by a skeleton. When an animator specifies an animation sequence, he defines the motion using this skeleton.

A skeleton is a connected set of segments, corresponding to limbs and joints. A joint is the intersection of two segments, which means it is a skeleton point where the limb

linked to that point may move. Motion control methods (MCMs) specify how an actor is animated and may be characterized according to the type of information it privileges when animating the Virtual Human (Thalmann and Thalmann, 1991). For example, in a keyframe system for an articulated body, the privileged information to be manipulated is the angle. In a forward dynamics-based system, the privileged information is a set of forces and torques; of course, in solving the dynamic equations, joint angles are also obtained in such a system, but they are considered as derived information. In fact, any MCM eventually has to deal with geometric information (typically joint angles), but only geometric MCMs explicitly privilege this information at the level of animation control. The nature of privileged information for the motion control of actors falls into three categories: geometric, physical and behavioral, giving rise to three corresponding categories of MCMs. Once the motion of the skeleton is designed, the realism of motion needs to be improved not only from the joint point-of-view, but also in relation to the deformations of bodies during animation. The body's inherent complexity makes things very difficult: A great many different materials that have no homogeneous behavior, from bones to muscles to fat tissues, come into play.

Since the overall appearance of a human body is very much influenced by its internal muscle structures, the layered model is the most promising for realistic human animation. The key advantage of the layered methodology is that once the layered character is constructed, only the underlying skeleton need to be scripted for animation; consistent yet expressive shape deformations are generated automatically.

Our model (Thalmann et al., 1996) is depicted by Figure 6. It incorporates a highly effective multi-layered approach for constructing and animating realistic human bodies.

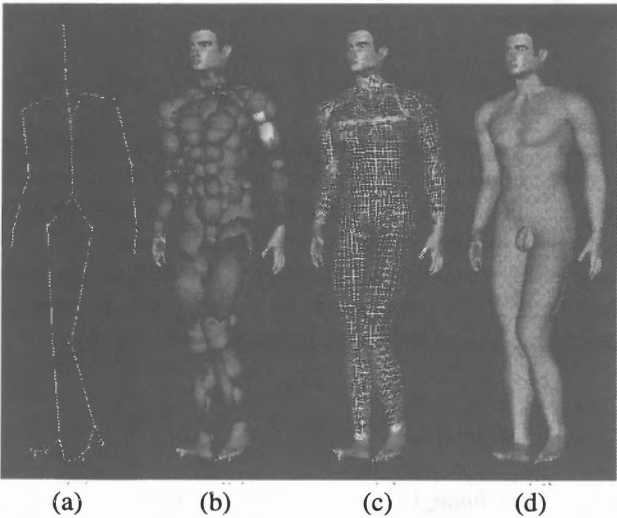(a)   (b)   (c)   (d)

Figure 6: The layered human body model: (a) Skeleton, (b) Ellipsoidal metaballs used to simulate muscles and fat tissue, (c) Polygonal surface representation of the skin, (d) Shaded rendering

Ellipsoidal metaballs are used to simulate the gross behavior of bone, muscle, and fat tissue; they are attached to the skeleton and arranged in an anatomically-based approximation. The skin construction is made in a three step process. First, the implicit surface resulting from the combination of the metaball's influence is automatically sampled along cross-sections with a ray casting method (Shen and Thalmann, 1995, Thalmann et. al., 1996). Second, the sampled points constitute 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 different skin pieces together and final rendering. The method, simple and intuitive, combines the advantages of implicit, parametric and polygonal surface representation, producing very realistic and robust body deformations. By applying smooth blending twice (metaball potential field blending and B-spline basis blending), the model's data size is significantly reduced.

4.2 Simplified Model of a Limb

To reduce the number of degrees of freedom (DOFs) and to be able to robustly estimate the skeleton's position, we replace the multiple metaballs of Section 4.1 by one ellipsoid attached to each bone in the skeleton, as shown in Figure 7(a).
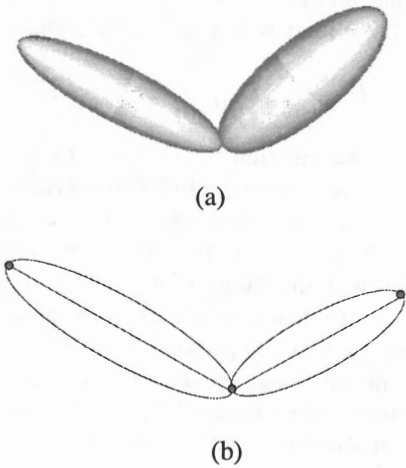

(a)

(b)

Figure 7: Simplified arm model. (a) Shaded view of the two ellipsoids representing the upperarm and the forearm. (b) Position of the two ellipsoids on the skeleton

4.2.1 Modeling the skeleton

The state of the skeleton is described by the state vector

$$S_{body}=[S_{skel}, S_{motion}] . \qquad (1)$$

The initial state of the skeleton S_{skel} consists of the rotations and translations from each DOF to the preceding one. It is fixed for a given body model. The variable state vector

S_{motion} contains the actual values for each DOF, i.e. the angle around the z-axis towards the next DOF. They reflect the position of the body with respect to its rest position. Thus, for any given joint, this state vector can be written as $S_{part}=[S_{pre},\Theta_i]$, where S_{pre} is the state vector for the preceding joint, and Θ_i is a rotation angle around the z-axis of that joint.

The joint local referential and coordinates are defined by a transformation matrix from a global referential to the local one. This matrix is computed recursively by multiplying all the transformation matrices that correspond to the preceding joints in the body hierarchy:

$$X_l = \prod_i D_i(S) \cdot X_\omega \quad ,$$

with $X_{l,\omega} = [x, y, z]^T$ being local, resp. global (world) coordinates and the homogeneous transformation matrices D_i, which depend on the state vector S, ranging from the root joint's first to the reference joint's last DOF. These matrices are of the form:

$$D = (RX + T) = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

The joints consist of several DOFs, each having its own transformation matrix $D = D_{rot_z} \cdot D_{ini}$. Take as example the elbow joint which has the two DOFs flex and twist:

$$D_{elbow} = D_{rot_{twist}} \cdot D_{ini_{twist}} \cdot D_{rot_{flex}} \cdot D_{ini_{flex}} \quad .$$

The "initial transformation" $D_{ini} = (RX + qT)$ is a matrix directly taken from the BODYlib skeleton. It translates by the bone length and rotates the local coordinate system from the previous to this DOF. The matrix entries are calculated with the values of the state vector S_{skel} and the variable coefficient q is necessary because we don't know the exact size of the person's limbs yet. For the first DOF of a joint this matrix is usually dense but the other DOFs have no translation ($T = [0,0,0]^T$) and the rotational part usually consists only of a swap of the axes to ensure that the DOF rotates around the z-axis:

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} .$$

The rotation matrix D_{rot_z} is a sparse matrix allowing only a rotation around the local z-axis (Θ_κ):

$$D_{rot_z} = \begin{bmatrix} \sin(\Theta_\kappa) & \cos(\Theta_\kappa) & 0 & 0 \\ \cos(\Theta_\kappa) & -\sin(\Theta_\kappa) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

4.2.2 Modeling the ellipsoids

The ellipsoids attached to the skeleton have a fixed position and orientation with respect to their enclosing joints and are assumed to be cylindrically symmetric around the longest axis. Their center lies in the middle of the bone and their axes coincide with the axis of the reference joint's local coordinate system. The corresponding positions are depicted by Figure 7(b). The origin and the angles of each ellipsoid are calculated in an incremental manner, since the position and orientation of parts which are further down the hierarchy tree depend on the positions and orientations of all previous joints. For example the forearm depends on the upper arm which depends on the shoulder and so on, until the root of the hierarchy is reached. Due to this incremental parameter calculation, the actual number of parameters for each body part differs.

We have chosen ellipsoids because, along with cylinders, they are the 3-D shapes with the least number of parameters (2: length and thickness plus the values of the skeleton's DOFs) that can be used to model human extremities. Ellipsoids, however, approximate more closely human extremities than cylinders. Furthermore, we rely on the rigid skeleton structure of Section 4.1 to constrain the length and connectivity of body parts. The different body parts are segmented before the optimization starts and we need not wait for a motion of the person to split a limb such as the arm into two parts, upperarm and forearm, as is the case in the work of (Kakadiaris and Metaxas, 1994).

More sophisticated models that include both global and local deformations, such as tapered superquadrics or Sethian's evolving surfaces (Malladi et al., 1995), may be able to approximate more closely the exact shape of the limb. However, they require the setting of more parameters and are thus harder to fit.

We represent 3-D ellipsoids using the standard implicit formulation:

$$\left(\frac{x_l - x_c}{r_x}\right)^2 + \left(\frac{y_l - y_c}{r_y}\right)^2 + \left(\frac{z_l - z_c}{r_z}\right)^2 = 1 \qquad (2)$$

where x_l, y_l and z_l are expressed in joint local coordinates of the bone to which the ellipsoid is attached and where $[x_c,y_c,z_c]$ denote its center and r_x, r_y, r_z its radii. The z axis is taken to be the one that is parallel to the bone.

In practice, we constrain the center of the ellipsoid to lie in the center of the bone and to be cylindrically symmetric around the axis of the bone. This can be written as:

$$x_c = y_c = 0$$
$$z_c = bone\_l$$
$$r_x = r_y = part\_\omega$$
$$r_z = bone\_l$$

where $bone\_l$ is half the bone length and $part\_\omega$ half the width of the body part (or thickness). Equation (2) can thus be rewritten as:

$$\left(\frac{x_l}{part\_\omega}\right)^2 + \left(\frac{y_l}{part\_\omega}\right)^2 + \left(\frac{z_l - bone\_l}{bone\_l}\right)^2 = 1 \quad (3)$$

where *bone_l* and *part_ω*, in addition to the values of the joint angles of Section 4.2.1, become the unknown to be adjusted by the optimization process of Section 5. In this way, we do not need an exact model of the specific person to capture his motion. Instead, we recover the model dimensions and its motion during one single processing step. We start from a "standard" body model and refine it during the fitting process to correspond as closely as possible to the person.
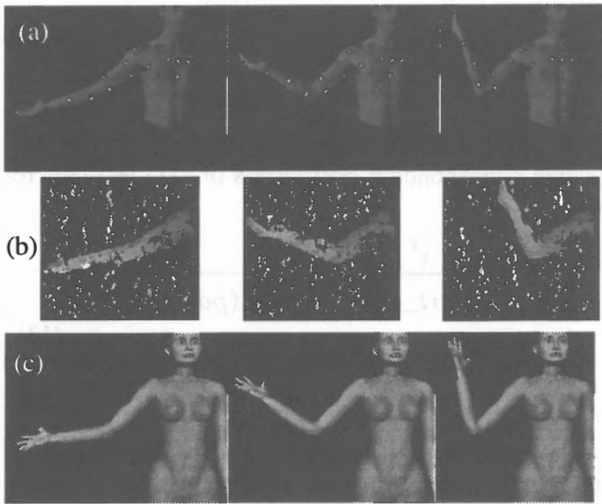
5. FITTING THE MODELS TO IMAGE DATA

Figure 8: Arm motion sequence: (a) The three left images of the three stereo pairs . (b) The corresponding disparity maps. (c) After motion recovery, a virtual human performs the same actions as the real person.

From a fitting point of view, the body model of section 4.2 embodies a rough knowledge about the shape of the body and can be used to constrain the search space. Our goal is to fix its degrees of freedom so that it conforms as faithfully as possible to the image data.

Here we use motion sequences such as the one shown in Figure 8 and corresponding stereo data computed using correlation based stereo (Fua, 1997). Thus, the expected output of our system is a state vector that describes the shape of the ellipsoids and a set of joint angles corresponding to their positions.

In this section, we introduce the least squares framework we use and show how we can exploit the tracking, stereo and silhouette data that we derive from the images.

5.1 Least Squares Framework

In standard least-squares fashion, we will use the image

data to write *nobs* observation equations of the form

$$f_i(S_{body}) = obs_i - \varepsilon_i \quad , \quad 1 \leq i \leq nobs \quad (4)$$

where S_{body} is the state vector of Equation 1 that defines the shape and position of the limb and ε_i is the deviation from the model. We will then minimize

$$v^T P v \Rightarrow Min \quad (5)$$

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 observations equations of Equation (4) as

$$f_i^{type}(S_{body}) = obs_i^{type} - \varepsilon_i^{type} \quad , \quad 1 \leq i \leq nobs \quad , \quad (6)$$

with weight P_{type}, where *type* is one of the possible types of observations we use. In this paper, *type* may be object space coordinates, silhouette position or other feature location information.

The individual weights of the different types of observations have to be homogenized prior to estimation according to:

$$\frac{p_i^k}{p_j^l} = \frac{(\sigma_j^l)^2}{(\sigma_i^k)^2} \quad , \quad (7)$$

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

$$\sum_{type=1}^{nt} v^{type^T} P_{type} v^{type} \Rightarrow Min \quad , \quad (8)$$

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

Since our overall problem is non-linear, the results are obtained through an iteration process.

5.2 Integrating Stereo Data

Let us assume that we are given a 3-D point that has been computed using stereo data. We want to minimize the distance of the reconstructed limb to all such "attractor" points. Given the implicit description of our ellipsoids, the simplest way to achieve this result for a single ellipsoid is to write an observation equation of the form:

$$\left(\frac{x^a}{part\_\omega}\right)^2 + \left(\frac{y^a}{part\_\omega}\right)^2 + \left(\frac{z^a - bone\_l}{bone\_l}\right)^2 = 1 - \varepsilon \quad , \quad (9)$$

871

where *bone_l* and *part_ω* are the state variables introduced in Section 4.2.2 and $[x^a, y^a, z^a]$ are the coordinates of the attractor point, expressed in local joint coordinates.

Equation (9) is actually a simplification of the stricter least squares modeling of this problem in form of condition equations with unknown parameters as

$$\left(\frac{x^a - \varepsilon_x}{part\_\omega}\right)^2 + \left(\frac{y^a - \varepsilon_y}{part\_\omega}\right)^2 + \left(\frac{z^a - \varepsilon_z - bone\_l}{bone\_l}\right)^2 = 1 \ . \quad (10)$$

In order to stay within the least squares framework as presented in chapter 5.1 we use the simplified observation equations (9) instead of (10).
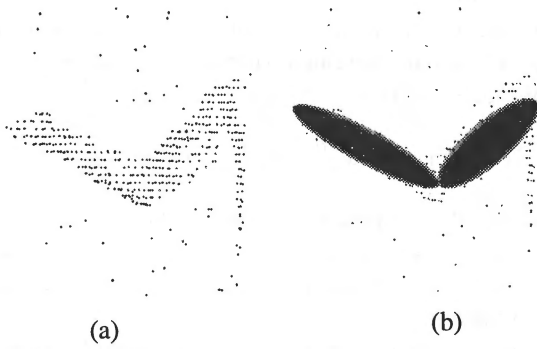

(a)   (b)

Figure 9: Fitting the model to cloud of points: (a) The cloud of points computed by one of the disparity maps of Figure 8(b). Note the many outliers. (b) The corresponding fit.

Since our limbs are made of more than one ellipsoid, for each attractor, we must also decide for which we should write Equation (9). In other words, we should decide to which body part we should attach the attractors. Here, because we use a small number of ellipsoids, we can simply evaluate the observation value using the formula of Equation (9) for each one and, at every iteration, pick the one that yields the smallest value. Once the body limbs are initialized there should be no problem assigning the attractors correctly through the motion sequence.

Because some of the attractors derived from stereo may be spurious, we use a variant of the Iterative Reweighted Least Squares technique (Beaton and Turkey, 1974, Fua, 1997) to discard outliers. We first fit the model by giving equal weight to all these attractors. We then weigh them inversely proportionally to their respective residuals and perform the optimization again. We iterate this process, which corresponds to the concept of robust estimation, several times.

Using this approach and the noisy stereo data of Figure 8(b), we can reconstruct the positions and shapes depicted by Figure 9. The joint angles stored in the state vector can then be used to animate the virtual human of Figure 8(c).

The prominent trajectory points obtained from tracking are introduced in the same manner as the attractor points from stereo above. However, due to their superior accuracy, they have assigned much higher weights.

5.3 Integrating Silhouette Data

Contrary to 3-D edges, silhouette edges are typically 2-D features since they depend on the viewpoint and cannot be matched across images. However, they constrain the surface tangent. Each point of the silhouette edge defines a line, the camera ray, that goes through the optical center of the camera and is tangent to the surface at its point of contact with the surface. The points of a silhouette edge therefore define a ruled surface that is tangent to the surface.

In terms of our model fitting this meas that the tangent plane for each silhouette ray may be formulated as

$$Ax + By + Cz + D = 0 \ , \quad (11)$$

with the coefficients A, B, C, D derived from silhouette image data and the given sensor orientation. A tangent plane onto an ellipsoid can be represented as

$$\frac{xx^s}{(part\_\omega)^2} + \frac{yy^s}{(part\_\omega)^2} + \frac{(z - bone\_l)(z^s - bone\_l)}{(bone\_l)^2} = 1 \quad (12)$$

with $[x^s, y^s, z^s]$ being the silhouette point P^s in object space. Equating corresponding coefficients of (11) and (12) results in

$$A = \frac{x^s}{(part\_\omega)^2} \ , \quad B = \frac{y^s}{(part\_\omega)^2} \ ,$$

$$C = \frac{z^s - bone\_l}{(bone\_l)^2} \ , \quad D = -\frac{z^s}{bone\_l} \ . \quad (13)$$

If we consider the coefficients A, B, C as (derived) observations we can set up the observation equations

$$A - \varepsilon_A = \frac{x^s}{(part\_\omega)^2} \ ,$$

$$B - \varepsilon_B = \frac{y^s}{(part\_\omega)^2} \ , \quad (14)$$

$$C - \varepsilon_C = \frac{z^s - bone\_l}{(bone\_l)^2} \ .$$

Here we have to introduce for each set of 3 observations 3 new unknown parameters x^s, y^s, z^s.

In addition, we can formulate the observation equations for the image coordinates (x'^s, y'^s) of the silhouette ray, based on collinearity conditions, as

$$x'^s - \varepsilon_x = f^x(P^s, eo, io) \ ; \quad p_x$$

$$y'^s - \varepsilon_y = f^y(P^s, eo, io) \ ; \quad p_y \quad (15)$$

with $P^s = [\ x^s, y^s, z^s\ ]$ .... object space coordinates
of the silhouette point
eo .... vector of exterior orientation elements
ie .... vector of interior orientation elements
p_x, p_y...corresponding weights.

Equations (14) should be amended by the constraint

$$D = -\frac{z^s}{bone\_l} \ . \tag{16}$$

In order to stay within the framework of least squares observation equations we also express (16) as observation equation, but with an associated very large weight p_D for the pseudo-observation D:

$$D - \varepsilon_D = -\frac{z^s}{bone\_l} \ ; \qquad p_D \tag{17}$$

Equations (14), (15) and (17) will contribute to (8).

6. CONCLUSIONS

We have presented some basic ideas how we intend to use image data from video sequences for the modeling of a human body under motion. This clearly represents an initial report with some preliminary results, including 3-D determination and tracking of passive marker points, extraction of silhouette data, generation of surface models by image matching, and setting up a framework for joint least squares estimation.

We have outlined a technique that allows us to fit a simplified animation model to noisy image data with very limited manual intervention. Because this model is closely related to the complete model we apply to perform animation, these results can be used to initialize this complete model and further refine it using the same data.

The capability we intend to develop will be of great applicability in an area such as the generation of feature films for entertainment. Generating and animating sophisticated models requires a tremendous amount of manual labor. While this may be appropriate for big-budget one-off use, the mass market of television entertainment is much more cost-driven and would benefit greatly from using techniques such as those described above. Furthermore, there currently is an inherent limit to the complexity of the animation models: Realism requires complex models, that is, models that are controlled by large numbers of parameters. As this number increases, so does the difficulty of the task faced by the designer. Automating the process will help solve this problem and will allow an increase in realism while reducing the cost.

REFERENCES:

Beaton, A. E. and Turkey, J. W., 1974. The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. Technometrics, 16, pp. 147-185

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

Gruen, A. and Li, H., 1997. Semi-Automatic Linear Feature Extraction by Dynamic Programming and LSB-Snakes. Photogrammetric Engineering and Remote Sensing, 63(8), pp. 985-995

Kakadiaris, I., Metaxas, D. and Bajcsy, R., 1994. Active Part-decomposition, Shape and Motion Estimation of Articulated Objects: A physics-based approach. Conference on Computer Vision and Pattern Recognition, 1994, pp. 980-984

Maas, H.-G., Gruen, A. and Papantoniou, D., 1992. Particle tracking velocimetry in three-dimensional flows - part I: Photogrammetric determination of particle coordinates. Experiments in Fluids 15, pp. 133-146

Maas, H.-G., 1998. Image sequence based automatic multi-camera system calibration techniques. International Symposium of ISPRS Commission V on "Real-Time Imaging and Dynamic Analysis", Hakodate, Japan, June 2-5

Malik, N., Dracos, T. and Papantoniou, D., 1992. Particle tracking velocimetry in three-dimensional flows - part II: Particle tracking. Experiments in Fluids 15, pp. 279-294

Malladi, R., Sethian, J.A., and Vemuri, B.C., 1995. Shape modeling with front propagation: A level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(2), pp. 158-175

Mortensen, E. N. and Barett, W. A., 1995. Intelligent Scissors for Image Composition. Computer Graphics, SIGGRAPH Proceedings, Los Angeles, pp. 191-198

Shen, J. and Thalmann, D., 1995. Interactive shape design using metaballs and splines. Implicit Surfaces, April

Thalmann, N.M. and Thalmann, D., 1991. Complex Models for Animating Synthetic Actors. Computer Graphics and Applications, pp. 32-44

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

Vaillant, R. and Faugeras, O. D., 1992. Using Occluding Conturs for 3D Object Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, February