International Archives of Photogrammetry and Remote Sensing,
Onuma, Japan, 1999, Vol. 32, Part 5-3W12, pp 133-140

HUMAN BODY MODELING FROM VIDEO SEQUENCES

Nicola D'Apuzzo

Institute of Geodesy and Photogrammetry (IGP)
Swiss Federal Institute of Technology Zurich (ETHZ)
ETH Hönggerberg, 8093 Zürich
SWITZERLAND
E-mail: nicola@geod.ethz.ch

Ralf Plänkers

Computer Graphics Lab (LIG)
Swiss Federal Institute of Technology Lausanne (EPFL)
1015 Lausanne
SWITZERLAND
E-mail: Ralf.Plaenkers@epfl.ch

KEY WORDS: Body modeling, Tracking, Video sequences

ABSTRACT

Synthetic modeling of human bodies and the simulation of motion is a long standing 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. Our ultimate goal is to automate the process and to produce realistic animation models given a set of video sequences.
In this paper, we show that, given video sequences of a person moving in front of a set of cameras, we can recover shape information and joint locations. Three synchronized cameras are used to acquire a sequence of triplets. 3-D data is extracted from the images in form of 3-D shape information and 3-D surface trajectories. We then outline techniques for fitting a simplified animation model to the 3-D data. The recovered shape and motion parameters can be used to either reconstruct the original sequence or to allow other animation models to mimic the subject's actions.

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. Our ultimate goal is to automate the process and to produce realistic animation models given a set of video sequences. 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.
In this paper, we show that, given stereo video sequences of a person moving in front of the camera, we can recover shape information and joint locations, both of which are essential to instantiate the model. This is achieved with minimal human intervention: to initialize the process, the user simply clicks on the approximate location of a few key joints in one image triplet. The recovered shape and motion parameters can be used to reconstruct the original motion or to make other animation models mimic the subject's actions. We concentrate on a video based approach because of its comparatively low cost and good 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 around the object.
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 work, we use video sequences acquired with three synchronized cameras to extract tracking and stereo information.
Recently, techniques have been proposed (Kakadiaris and Metaxas, 1996, Gavrila and Davis, 1996, Lerasle et al., 1996, Davis and Bobick, 1998) to track human motions from video sequences. They are fairly effective but use very simplified models of the human body, such as ellipsoids, that do not precisely model the human shape and would not be sufficient for a truly realistic simulation.
Much work has also been devoted to the use of silhouettes for body modeling (Davis and Bobick, 1998, Hilton et al., 1999). They provide very useful but incomplete information about shape which is one of the issues we will address in this work. Here, we use stereo information to instantiate the sophisticated animation models that we have developed in the past to both track the motion and recover the shape of the body as accurately as possible. However, silhouette information can easily be integrated into our extensible least squares framework.
We first introduce our approach to computing 3-D stereo information and 3-D surface trajectories. We then present the animation model we use. Finally, we introduce our fitting procedure and show how we can handle the different kinds of input information.

2. EXTRACTING 3-D DATA

2.1 Image acquisition system

Three synchronized CCD cameras in a linear arrangement (left, center, right) are used (Figure 1).

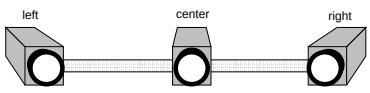

Fig. 1: Arrangement of the three CCD cameras

A sequence of triplet images is acquired with a frame grabber and the images are stored with 768x576 pixels at 8 bit quantisation. The CCD cameras are interlaced, i.e. a full frame is split into two field which are recorded and read out consecutively. As odd and even lines of an image are captured at different times, a saw pattern is created in the image when recording moving objects. For this reason only the odd lines of the images are processed, at the

cost of reducing the resolution in vertical direction by 50 percent. To calibrate the system, the reference bar method (Maas, 1998) is used. 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.
In this paper, we show the results achieved with our system analysing a walk sequence acquired at the LIG lab in Lausanne. Figure 2 shows part of the sequence, which is composed of 30 frames.

Fig. 2: 6 frames of the walk sequence (upper left to lower right)

2.2 Surface measurement

Our approach is based on multi-image photogrammetry. Three images are acquired simultaneously by three synchronized cameras. A multi-image matching process (D'Apuzzo, 1998) establishes correspondences in the three images starting from a few seed points. It is based on the adaptive least squares method (Gruen, 1985) which considers an image patch around a selected point. One image is used as template and the others as search images. The patches in the search image are modified by affine transformations (translation, rotation, sheering and scaling) and the grey levels are varied by multiplicative and additive constants. The algorithm finds the corresponding point in the neighbourhood of the selected point in the search images by minimizing the sum of the squares of the differences between the grey levels in these patches. Figure 3 shows the result of the least squares matching with an image patch of 13x13 pixels. The black box represents the patche selected (initial location in the search image) and the white box represents the affinely transformed patch in the search image.

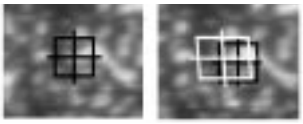

Fig. 3: Least squares matching algorithm (LSM)
Left: template image, right: search image

To define the seed points of the multi-image matching process, approximations for a few corresponding points are manually selected in the three images. For example, for the full body sequence of figure 2 we selected about 20 seed points manually. The least squares algorithm is applied to find their exact location in the pictures. To define the regions between the different seed points, a Voronoi tessellation in the template image is computed. The image is divided into polygonal regions according to which of the seed points is closest. Starting from the seed points, the stereo matcher automatically determines a dense set of correspondences. The central image is used as a template image and the other two (left and right) are used as search images. The matcher searches the corresponding points in the two search images independently. At the end of the process, the data sets are merged to become triplets of matched points. The matcher uses the following strategy: the process starts from one seed point, shifts horizontally in the template and in the search images and applies the least squares matching algorithm in the shifted location. If the quality of the match is good, the shift process continues horizontally until it reaches the region boundaries. The covering of the entire polygonal region of a seed point is achieved by subsequently horizontal and vertical shifts (Figure 4).

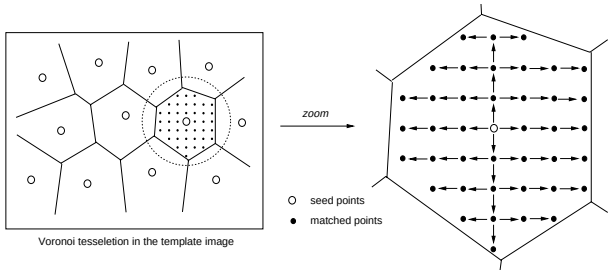

Fig. 4: Search strategy for the establishment of correspondences between images

If the quality of the match is not satisfactory, the algorithm works adaptively by changing parameters (e.g. smaller shift, bigger size of the patch). The search process is repeated for each seed point region until the whole image is covered. At the end of the process, holes of areas not analysed can appear in the set of matched points. The algorithm tries to close these holes by searching from all directions around. The matching process results in a set of matched points in the three images. To compute the 3-D coordinates of these points, we apply forward intersection using the orientation and calibration data of the cameras. Figure 5 shows the noisy results of the 3-D surface measurement of the first frame of the sequence shown in figure 2.

Fig. 5: Measured surface of the body in the first frame

2.3 Tracking process

The tracking process is also based on least squares matching techniques. The spatial correspondences between the three images of the different views and also the temporal correspondences between subsequent frames are computed using the same least squares matching algorithm mentioned before. To start the process a triplet of corresponding points in the three images is needed. This 3-D point is then tracked through the sequence in the three images and therefore its 3-D trajectory can be computed. Figure 6 shows the use of the least squares matching algorithm to track the point.

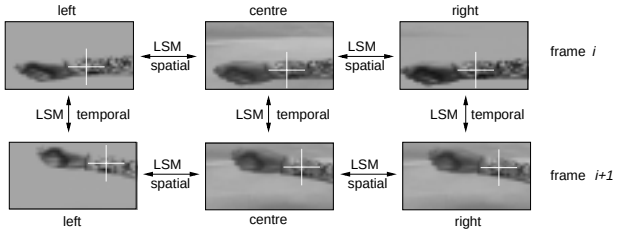

Fig. 6: Tracking process

In frame *i*, a triplet of corresponding points in the three images is established with the least squares matching algorithm (*spatial LSM*). In each of the three images (left, center, right) a corresponding point is matched in the next frame *i+1* also with the least squares matching algorithm (*temporal LSM*). Figure 7 depicts how the temporal correspondences are established between subsequent frames.

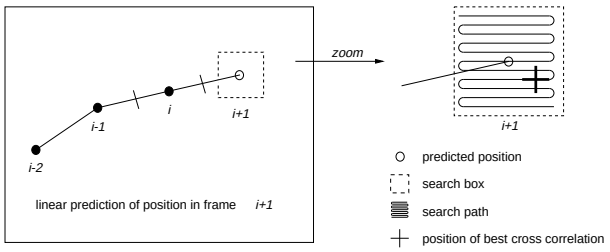

Fig. 7: Tracking in image space: temporal LSM is applied at the position of best cross correlation

For frame *i+1*, a linear prediction of the position of the tracked point from the previous frame is made. A search box is defined around this predicted position in the frame *i+1*. This box is scanned for searching the position which has the best value of cross correlation between the image of frame *i* and the image of frame *i+1*. This position is considered an approximation of the exact position of the point to be tracked. The least squares matching algorithm is applied at that position and the result can be considered the exact position of the tracked point in the new frame. This process is performed independently for the three images of the different views. A *spatial LSM* is executed at the positions resulting from the *temporal LSMs* and if no significant differences occur between the two matches, the point can be considered exactly tracked. The tracked point's 3-D trajectory is determined by computing the 3-D coordinates of the point through the sequence by forward intersection. Velocities and accelerations are also computed.
Figures 8 shows some tracked points through the walk sequence and the figure 9 shows its computed 3-D trajectories.

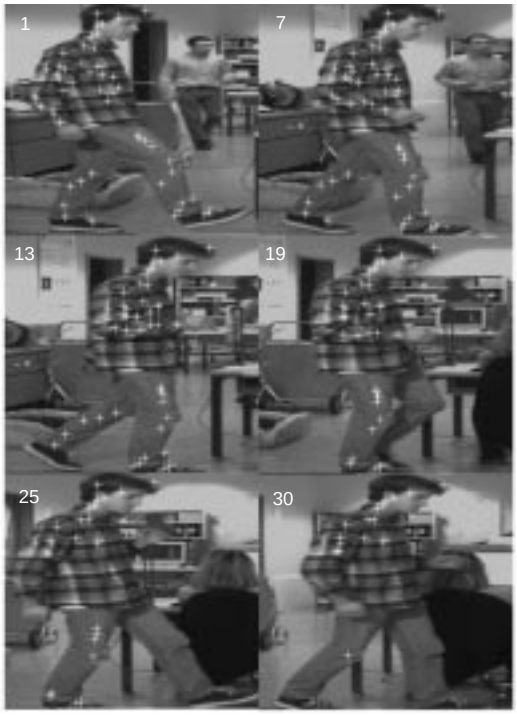

Fig. 8: Tracking few points (remark: to avoid the interlace effect, only the odd lines of the images are processed; their size is therefore halved vertically)

Fig. 9: Computed 3-D trajectories of the tracked points left: frontal view, right: lateral view

Tracking single points in this way may produce errors which cannot be easily detected. The only control of the tracking result is the LSM test, there is no 3-D control of the trajectories. Thus, false trajectories can be generated even if the tracking results seems good. This can happen in cases of fast movement and poor texture of the measured surface.
To solve this problem, the tracking algorithm is applied to all the points measured on the surface of the first frame (Figure 10).

Fig. 10: Tracking all points measured in the first frame left: frontal view, right: lateral view

The result can be seen as a vector field of trajectories (position, velocity and acceleration), so that at the end, the results can be checked for consistency and local uniformity of the movement. Two filters are applied to the results to remove or truncate false trajectories (Figure 11).

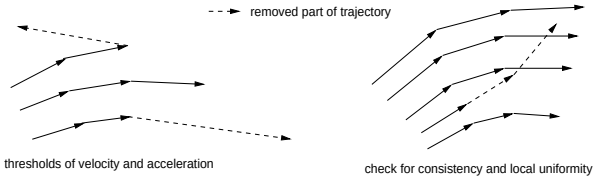

Fig. 11: Filters to remove false trajectories

The first filter consists of thresholds for the velocity and acceleration. The second filter checks for the local uniformity of the motion, both in space and time. Since the human body can be considered as an articulated moving object, the resulting vector field of trajectories must be locally uniform, i.e. the velocity vector must be nearly constant in sufficiently small regions at a particular time. To check this property, the single trajectories are compared to local (in space and time) mean values of the velocity vector (figure 12). If the differences are too large, the trajectory is considered to be false and it is truncated or removed.

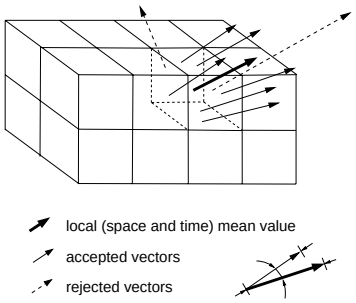

Fig. 12: Consistency and uniformity filter.
Space is divided in local regions. At each frame, the vectors in the local region are compared with the local mean value

As can be seen comparing figure 13 with figure 12, almost all the false trajectories are removed or truncated by the two filters.

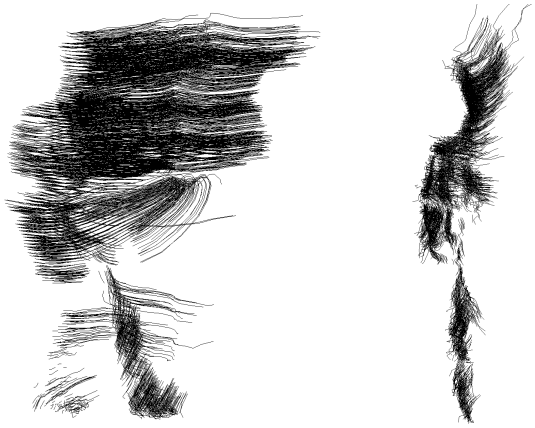

Fig. 13: 3-D trajectories after the filtering
left: frontal view, right: lateral view

Figure 14 shows, the tracking points in image space. The set of points constantly becomes smaller during the process, because erroneous trajectories are removed or truncated by the filters and because points disappear from the scene and new points appear in the scene during the sequence.

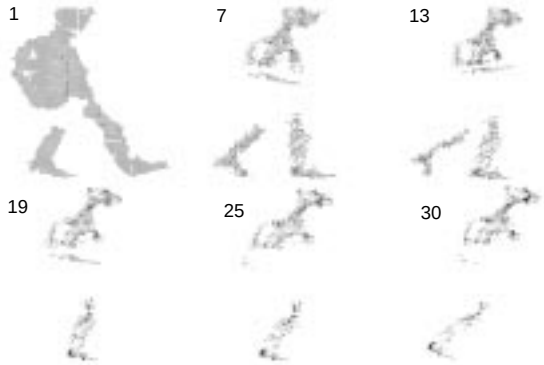

Fig. 14: Tracked points displayed in image space

An important improvement of the process is an automatic generation of new points to track, which can be either new appearing points or points previously lost during the tracking process.

3. 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 fitted to noisy data without an 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.

3.1 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.
Our model (Thalmann et al., 1996) is depicted by Figure 15. It incorporates an highly effective multi layered approach for constructing and animating realistic human bodies.

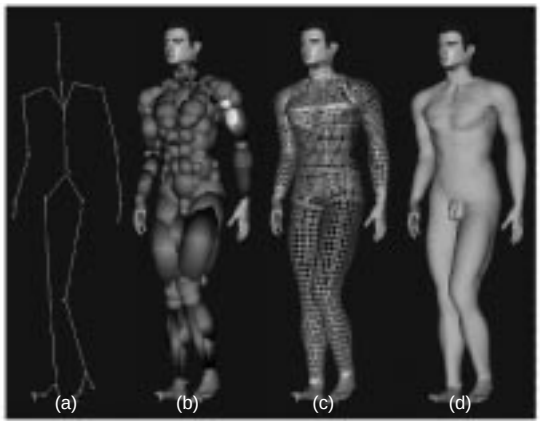

Figure 15: The layered human body model: (a) Skeleton,
(b) Ellipsoidal metaballs used to simulate muscle and fat tissue,
(c) Polygonal surface representation of the skin,
(d) Shaded rendering

Ellipsoidal metaballs are used to simulate the gross behaviour 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 metaballs 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.
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.

3.2 Skeleton and state vector

The state of the skeleton is described by the state vector

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

Since the skeleton is modeled in an hierarchical manner, we can define the *static* or *init* state of the skeleton S_{skel} as the rotations and translations from each joint with respect to the preceding one. It is fixed for a given instance of the body model. The variable or *motion* state vector S_{motion} contains the actual values for each degree of freedom (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. All joints have a single angular DoF. More complicated articulations are split into several, single DoF joints sharing the same location and only differing in their orientation.
The position of joints in a global or world referential is obtained by multiplying the local coordinates by a transformation matrix. This matrix is computed recursively by multiplying all the transformation matrices that correspond to the preceding joints in the body hierarchy:

$$X_l = \prod_i D_i(S) \cdot X_w \ , \qquad (2)$$

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

$$D = D_{rot_z} \cdot D_{ini} \ . \qquad (3)$$

The rotation matrix D_{rot_z} is defined by the motion state vector. It is a parse matrix allowing only a rotation around the local z-axis (Θ_κ). The static transformation $D_{ini} = (RX + sT)$ is a matrix directly taken from the standard skeleton. These matrices translate by the bone length and rotate the local coordinate system from the joint to its parent. The matrix entries are calculated with the values *s* of the state vector S_{skel}. The variable coefficient *s* is necessary because the exact size of the limbs may vary from person to person.

3.3 Simplified model of a limb

To robustly estimate the skeleton's position and to reduce the number of DoFs, we replace the multiple metaballs of Section 3.1 by only three metaballs attached to each limb (figure 16).

Fig. 16: Simplified model for fitting. Although the metaballs are displayed as distinct ellipsoids, they blend into each other to form a single smooth surface

In an earlier approach (Fua et al., 1998a, Fua et al., 1998b), we used only one ellipsoid per limb. This had the advantage of being fast to compute but the errors introduced by the model's imperfection were large enough to lead to unsatisfactory fittings. We therefore decided to use a slightly more complicated model which approximates better the shape of human limbs. To reduce the number of DoFs we introduced higher level parameters which cover a number of direct metaball parameters. Like upper arm width which controls the relative size of all metaballs in the region of the upper arm.
The metaballs are rigidly attached to the skeleton. They have a fixed orientation and a fixed position relative to the length of the limb. Only their size, i.e. their radii, are subject to modification by the fitting process.
The different body parts are segmented before the fitting starts. This is simply done during the initialization phase where the model takes an approximate posture which is good enough to assign a 3-D observation to the closest limb. Thus, we do not have to wait for a motion of the person to split a limb such as the arm into two parts, upper arm and forearm. The segmentation is reversible as it is redone after several iterations and, thus, possible segmentation errors due to a wrong initialization are removed during the fitting process.

3.4 Metaballs and their mathematical description

3.4.1 Definition

In Blinn's basic formulation (Blinn, 1982), *metaballs* or *blobs* are defined by a set of points $P_i(x_i,y_i,z_i)$ that are the sources of a potential field. Each source is defined by a *field function* $F_i(x,y,z)$ that maps $\Re^3$ to $\Re$, or a subset of $\Re$. At a given point $P(x,y,z)$ of the Euclidean space, the fields of all sources are computed and added together, leading to the global field function $F(x,y,z)$:

$$F(x, y, z)= \sum_{i=1}^{n} F_i(x, y, z) \qquad (4)$$

A curved surface can then be defined from the global field function *F* by giving a threshold value *T* and rendering the following equipotential surface ***S*** for this threshold:

$$\boldsymbol{S} = \{(x, y, z) \in \Re^3 \,|\, F(x, y, z) = T\} \qquad (5)$$

Conceptually it is usually simpler to consider field function F_i as the composition of two functions (Blanc and Schlick, 1995): the *distance function* d_i which maps $\Re^3$ to $\Re^+$, and the *potential function* f_i which maps $\Re^+$ to $\Re$:

$$F(x, y, z) = \sum_{i=1}^{n} f_i(d_i(x, y, z)) \quad . \qquad (6)$$

The function $f_i(d)$ characterizes the distance between a given point $P(x,y,z)$ and the source point $P_i(x_i,y_i,z_i)$. Typically d_i is defined as a function of a user provided parameter $r_a \in \Re^+$ (called *effective radius*) which expresses the growing speed of the distance function. The most obvious solution for $d_i(x,y,z)$ is the euclidean distance, but several other functions have been proposed in the literature, especially when the potential source is not reduced to a single point or its field is not equally distributed in space.

3.4.2 Distance function

In this work, we only consider ellipsoids as primitives because they are relatively simple but, nevertheless, allow modeling of human limbs with a fairly low number of primitives and thus number of parameters. We represent the distance function d_i by the implicit distance to the ellipsoid that is

$$d_i(x, y, z) = \left(\frac{x}{l_x}\right)^2 + \left(\frac{y}{l_y}\right)^2 + \left(\frac{z}{l_z}\right)^2 \quad , \qquad (7)$$

where $L_i = (l_x,l_y,l_z)$ are the radii of the ellipsoid, i.e. half the axis length along the principal directions.

3.4.2 Potential function

The field value at any point P in space is defined by the distances between P and the source points P_i. The center of the primitive, its source, has the greatest density. The value of the primitive's density, or *weight*, decreases toward the element's outer edge, or effective radius.
The visible size of a primitive, called the *threshold radius*, is determined by the effective radius and weight. Field functions should satisfy two criteria:

1. Extremum: The contribution at the source is some maximum value w_0, and the field will drop smoothly to zero at a distance r_a, the effective radius.

2. Smoothness: In order to blend multiple metaballs smoothly and gradually, $f'(0)=f'(r_a)=0$.

A single, lower degree polynomial cannot meet both criteria, hence either piece wise quadratic or high order polynomials have been proposed. Their disadvantage are a high complexity and thus high computational cost.
Here we are attempting to fit the model to 3-D data by minimizing an objective function. In order to do so, we need to work on a well defined mathematical basis and the smoothness criterion is essential when fitting a shape with multiple metaballs. We therefore use an exponential field function:

$$f_i = w_i \cdot \left(\frac{1}{e^d}\right)^2 = w_i \cdot exp(-2d) \quad , \qquad (8)$$

with d being defined as in equation 7 and the weight being fixed for the moment ($w_0=1$, $w_t=0.5$). In the future, we might leave the weight as a free parameter for the fitting since it allows to easily model sharper edges.
An exponential field function is also more effective in the least squares fitting framework because its derivatives are very easy to compute. Its equipotential surface $\boldsymbol{S}$ is only slightly different from the standard representation and, more importantly, it never falls to zero.
This last property has two consequences:

1. Each blob has an influence on all other blobs of the same limb, although, it will become very small for distant blobs. This is obviously undesired for modeling purposes since the designer looses local control.

2. At the same time as each blob influences all other blobs, each blob is influenced by all observations in our fitting framework. This allows us to work with only a rough initialization of the model's posture because of the long range effect of the *exp()* function. Since the observations are already segmented and associated to body parts, the unlimited influence does not pose any problems on the other body parts.

4. FITTING THE MODELS TO IMAGE DATA

From a fitting point of view, the body model of section 3.3 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 2 and corresponding stereo data computed using the method of section 2.2. Thus, the expected output of our system is a state vector that describes the shape of the metaballs and a set of joint angles corresponding to their positions in each frame.
In this section, we introduce the least squares framework we use and show how we can exploit the tracking and stereo and data that we derive from the images.

4.1 Least squares framework

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

$$f_i(S) = obs_i - \varepsilon_i \quad , \qquad 1 \le i \le nobs \qquad (9)$$

where S 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

$$v^T P\, v \Rightarrow Min \qquad (10)$$

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 9 as

$$f_i^{type}(S) = obs_i^{type} - \varepsilon_i^{type} \quad , \quad 1 \le i \le nobs \quad , \qquad (11)$$

with weight P_i^{type}, where *type* is one of the possible types of observations we use. In this paper, *type* is restricted to object space coordinates, although other information cues can easily be integrated.
The individual weights of the different types of observations have to be homogenized prior to estimation according to:

$$\frac{p_i^k}{p_j^l} = \frac{(\sigma_j^l)^2}{(\sigma_i^k)^2} \quad , \tag{12}$$

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

$$\sum_{type\,=\,1}^{nt} v^{type^T} P_{type} v^{type} \Rightarrow Min \quad , \tag{13}$$

with *nt* the 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. We use a modified version of the Levenberg-Marquardt algorithm (Press et al., 1986) which is able to deal with the huge number of observations we encounter.

4.2 Using tracking data

The 3-D tracking information of section 2.3 serves to capture robust stereo information and to initialize the body model in all frames. The algorithm is initialized by letting the user specify an approximate posture and position of the model in the first frame of the sequence. The results of the tracking deliver the approximate positions of the visible articulations for the rest of the sequence.

4.3 Using stereo data

3-D points such as the ones computed with the technique of section 2.2 or any other source of 3-D information can be used. We want to minimize the distance of the reconstructed limb to all such "attractor" points. Given the implicit description of our metaballs, the simplest way to achieve this result is to write a pseudo observation equation of the form:

$$\sum_{i\,=\,1}^{np} w_i \cdot \left(\frac{1}{e^{d_i}} \right)^2 = w_t - \varepsilon \tag{14}$$

$$\sum_{i\,=\,1}^{np} \left(\frac{1}{e^{\left(\frac{x_i}{lx_i}\right)^2 + \left(\frac{y_i}{ly_i}\right)^2 + \left(\frac{z_i}{lz_i}\right)^2}} \right)^2 = \frac{1}{2} - \varepsilon \quad , \tag{15}$$

where *np* is the number of primitives for this body part, $P_i(x_i,y_i,z_i)$ is the 3-D observation transformed into the local coordinates of primitive *i* with radii $L_i(lx_i,ly_i,lz_i)$. We use Equation 15 which is the same than Equation 14 except for the fixed weights $w_t=0.5$, $w_i=1$, $i \in [1,np]$.
The optimization is effected wrt. the primitives' radii L_i and the DoFs which reside in the transformation of each observation from world global to primitive local coordinates. These DoFs consist of the motion parameters and the skeleton parameters, i.e. length of each limb. According to equation 2, each P_i can be written as a function of its world coordinates and the elements of state Vector *S*. In practice, we experienced better convergence by iteratively alternating between primitive parameters and skeleton parameters instead of optimizing them simultaneously. For more detail we refer the interested reader to a previous publication (Fua et al., 1998a).

4.4 Preliminary results

Figure 17 shows the results of the fitting process for four frames at the beginning of the walk sequence of figure 2. The simpified model of section 3.3 is fitted to the 3-D data extracted from the images..

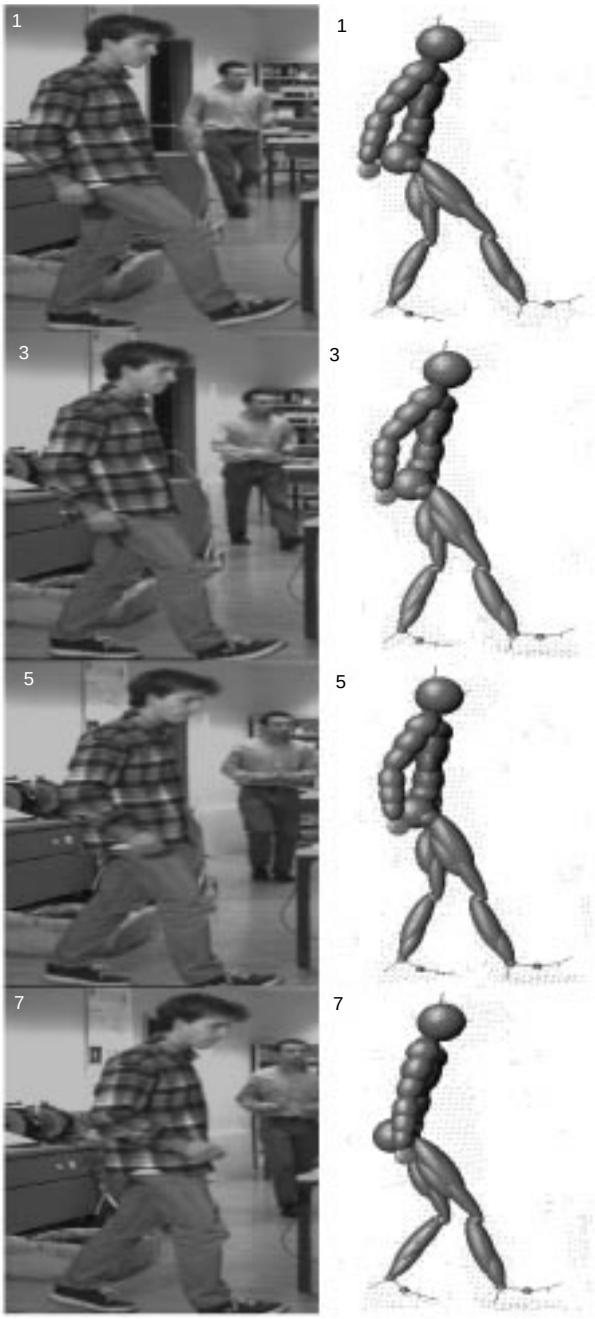

Fig. 17: Results of the fitting process for the beginning of the analysed sequence. Left: original frame, right: fitting results (remark: the images are centered to the body, i.e. the model looks stationary but in reality it moves to the right)

It can be seen that the legs follow very well the motion, whereas the body lags a bit behind. The arm does its best to keep track but since we don't use any prediction yet, the occluded body doesn't move enough. This results in a rather funny pose in the last frame. In future work, we will investigate the possibilities of having the model guide the tracking process. If a point on the body's surface vanishes due to occlusion, we can employ the model to predict where and when it will appear again. The implementation of this function will radically improve the quality of the results of the fitting process.

5. CONCLUSIONS

In this paper, we have shown that given video sequences of a moving person acquired with a multi camera system, we can recover shape information and track joint locations during the motion. We have outlined techniques for fitting a complete animation model to noisy stereo data and we have presented a tracking process based on least squares matching. The recovered shape and motion parameters can be used to create a realistic animation. Our ultimate goal is to produce automatically, with minimal human intervention, realistic animation models given a set of video sequences. The capability we intend to develop will be of great applicability in animation areas, since the techniques used nowadays require a very long time of manual work to generate and animate sophisticated models of humans. Automating the process will allow an increase of realism with simultaneous decrease of costs.

8. AKNOWLEDGMENTS

The work reported here was funded in part by the Swiss National Science Foundation.

REFERENCES:

Blanc C. and Schlick C., 1995. Extended field functions for soft objects. In Eurographics Workshop on Implicit Surface 95, pp. 21-32, Grenoble, France, 1995.

Blinn J. F., 1982. A generalization of algebraic surface drawing. ACM Transactions on Graphics, 1(3), pp. 235-256, 1982.

D'Apuzzo N., 1998. Automated photogrammetric measurement of human faces. International Archives of Photogrammetry and Remote Sensing, 32(B5), pp. 402-407, Hakodate, Japan, 1998

Davis J. and Bobick A., 1998. A Robust Human-Silhouette Extraction Technique for Interactive Virtual Environments. Workshop on Modelling and Motion Capture Techniques for Virtual Environments, pp. 12-25, Geneva, Switzerland, November 1998.

Fua P., Gruen A., Plänkers R., D'Apuzzo N. and D. Thalmann, 1998a. Human body modeling and motion analysis from video sequences. International Archives of Photogrammetry and Remote Sensing, volume 32(B5), pp. 866-873, Hakodate, Japan, 1998.

Fua P., Plänkers R. and D. Thalmann,1998b. From image synthesis to image analysis: Using human animation models to guide feature extraction. Fifth International Symposium on the 3-D Analysis of Human Movement, Chattanooga, TN, July 1998.

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

Gruen A., 1985. Adaptive least squares correlation: a powerful image matching technique. South African Journal of Photogrammetry, Remote Sensing and Cartography, 14(3), pp. 175-187, 1985.

Hilton A., Beresford D., Gentils T., Smith R. and Sun W., 1999. Virtual People: Capturing Human Models to Populate Virtual Worlds. Computer Animation, Geneva, Switzerland, May 1999.

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

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

Maas H.-G., 1998. Image sequence based automatic multi-camera system calibration techniques. International Archives of Photogrammetry and Remote Sensing, 32(B5), pp. 763-768, Hakodate, Japan, 1998.

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

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

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