Articulated Motion Capture from 3-D Points and Normals

Matti Niskanen, Edmond Boyer, Radu Horaud

Articulated motion capture from 3-D points and normals

Matti Niskanen*
Machine Vision Group
Infotech Oulu
University of Oulu
PO Box 4500, Finland

Edmond Boyer and Radu Horaud
Perception Group
INRIA Rhône-Alpes
655, avenue de l'Europe
38330 Montbonnot, France

Abstract

In this paper we address the problem of tracking the motion of articulated objects from their 2-D silhouettes gathered with several cameras. The vast majority of existing approaches relies on a single camera or on stereo. We describe a new method which requires at least two cameras. The method relies on (i) building 3-D observations (points and normals) from image silhouettes and on (ii) fitting an articulated object model to these observations by minimizing their discrepancies. The objective function sums up these discrepancies while it takes into account both the *scaled algebraic distance* from data points to the model surface and the *offset in orientation* between observed normals and model normals. The combination of a feed-forward reconstruction technique with a robust model-tracking method results in a reliable and efficient method for articulated motion capture.

1 Introduction

In this paper we address the problem of estimating the motion parameters of articulated objects, such as humans, from 3-D points and normals. These entities are inferred from 2-D silhouettes gathered with several synchronized cameras, Figure 1. The problem of tracking articulated shapes has been thoroughly studied in the recent past and a number of interesting methods and software packages are available. The vast majority of existing approaches and solutions relies on a single camera (a video sequence), on stereo (both binocular and trinocular), or on a large number of cameras. The first class of methods (a single video) attempts to recover the motion parameters directly from images and requires sophisticated probabilistic modelling. The second class of methods relies on depth data which, in turn, require search methods in order to solve for the stereo correspondence problem. The third class of methods relies on space-carving and level-set methods which are still under development. The latter has proved their usefulness for 3-D shape modelling but not for recovering motion parameters.

Here we describe a method which needs 2 to 6 cameras evenly distributed around the scene, i.e., they do not need to be arranged such that stereo correspondence is optimized. The method consists in fitting the pose of an articulated object model to 3-D observations gathered at some time instant, provided that the pose at the previous time instant has already been estimated. The object model is described by an *articulated implicit surface* that embeds a kinematic structure (such as a human body, a hand, an animal, etc.) and a set of volumetric primitives (ellipsoids).

*M. Niskanen is funded by Infotech Oulu. Financial support from Seppo Säynäjäkangas foundation is kindly aknowledged.

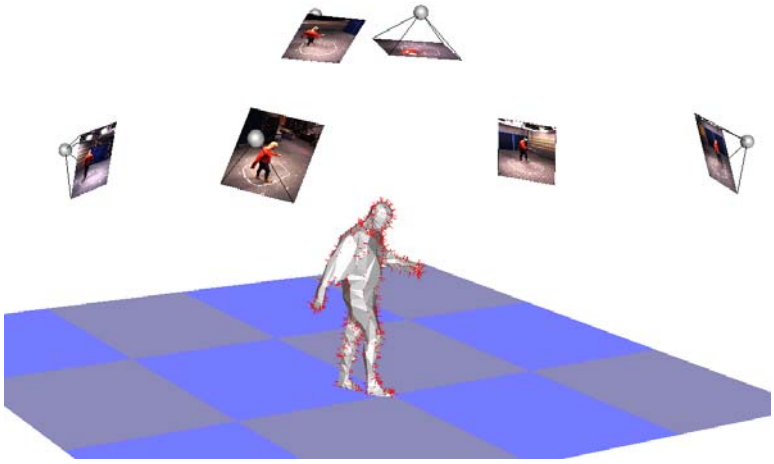

Figure 1: The cameras overlook a scene and a reconstruction method estimates 3-D points (connected to form a mesh for the purpose of the display) as well as 3-D vectors (shown as a needle field) normal to a smooth surface.

The implicit surface is defined as a distance function over these primitives and therefore this surface is simply a level set over a blending of these ellipsoids.

The 3-D observations are computed from image silhouettes gathered with the cameras. These 3-D data consist in surface patches, i.e., a 3-D point and a 3-vector. In order to fit these observations to the model we define a surface-patch-to-implicit-surface distance. The objective function to be minimized over the motion parameters is a sum of squares of the distances just mentioned.

Previous work. Since we adopt an "image understanding" point of view, we immediately rule out systems based on magnetic or optical markers, special-purpose clothes, and so forth. For a general review of human motion capture methods see [14]. Methods based on a single image sequence require a probabilistic framework [1], [8], and many others. An intrinsic difficulty, however, with methods based on 2-D data is the ambiguity of associating a multiple degree-of-freedom 3-D model with image contours, texture, and optical flow [4], [7]. Other researchers combine several cameras and make use of 2-D silhouettes whose image deformation is related, among others, to 3-D motion parameters. In [9], 2-D image data apply forces to a projected model and the parameters of the latter are adjusted such that the force field is minimized.

Methods using 3-D data are the most relevant with respect to our own approach. In general 3-D data are produced using stereo [5], [15], [6]. An articulated model based on cylindrical parts and an ICP algorithm is used in [5]. Both [15] and [6] use implicit surfaces defined over a set of spheroids, and these two methods are the most closely related to our own approach. In [15] an algebraic distance is minimized in order to fit the implicit surface to the depth data, and silhouette observations are used to constrain this surface to be tangent to rays originating at the optical center of the camera and passing through silhouette points. In [6] the stereo data are fitted to the model using an EM algorithm. Moreover, 3-D data that are consistent with the model are incrementally added to the latter such that both point-to-point and point-to-surface distance

errors contribute to the fitting.

Original contributions. This paper has the following original contributions: First, 3-D observations (both points and normals) are computed from 2-D silhouettes based on multiple-camera geometric constraints and on the hypothesis that the observed 3-D surfaces are locally smooth; The method may well be viewed as an improvement over convex hull computation. There is no need to arrange the cameras such that stereo matching performs in an optimal manner. Second, the objective function, measuring the discrepancy between model and data, takes into account both point-to-surface and data-normal-to-model-normal discrepancies. We derive an analytic expression for these discrepancies which allows the straightforward implementation of non-linear minimization techniques. Third, the method avoids image projections of complex models. Fourth, data-to-model fitting is achieved in a single 3-D metric space instead of multiple, possibly inconsistent, 2-D projective spaces.

Organization. The remainder of this paper is organized as follows. Section 2 describes how 3-D data are obtained from image silhouettes. Section 3 describes the articulated model which is based on zero-reference kinematic chains, on ellipsoids, and on an articulated implicit surface defined over these chain and volumetric primitives. Section 4 describes the fitting between the data and the model based on both points and surface normals. Section 5 describes results obtained with both simulated and real data. Finally Section 6 draws some conclusions and suggests directions for future work.

2 Surface patches from image silhouettes

In this section we describe how 3-D points and surface normals are inferred from multiple image silhouettes. The 3-D shape data that we estimate consist in the positions of points and normals associated with the 3-D surface that produced the silhouettes. Such shape information is closely related to the visual hull of an object and it shares with the latter its robustness. Nevertheless, it is richer than the visual hull alone since it includes not only the surface tangent planes but also the surface positions which are not given by the visual hull. To estimate these positions, we use the fact that our surface models, ellipsoids, are C^2 surfaces. The method is valid, more generally, for locally smooth surfaces of order 2.

Viewing edges. We assume that a set of silhouettes – that segment the input images into foreground and background – are provided. These silhouettes may be combined to give rise to a *visual hull* which is the maximal 3-D shape consistent with them. The visual hull does contain the body surface and may intuitively be seen as the intersection of the *viewing cones* associated with the silhouettes. *Viewing edges*, or bounding edges [10], are the intervals along the viewing lines, as shown on Figure 2. They correspond to viewing-line contributions to the visual-hull surface and therefore they are associated to image points lying onto the silhouette boundary curves. Computing such a set of viewing edges is fast, simple, well-defined, and has already been used in various reconstruction applications, [12] and [3].

A silhouette is described by a discrete set of 2-D points. Viewing edges along a viewing line may be defined by combining silhouettes from two images and the associated epipolar constraint, as depicted in Figure 2. This can be easily extended to an arbitrary number of images and silhouettes. Whenever an additional silhouette from a new image is available, the viewing edges are updated to be consistent with contributions from the additional silhouette points. As the number of silhouettes increases, the length of the viewing edges narrows down.

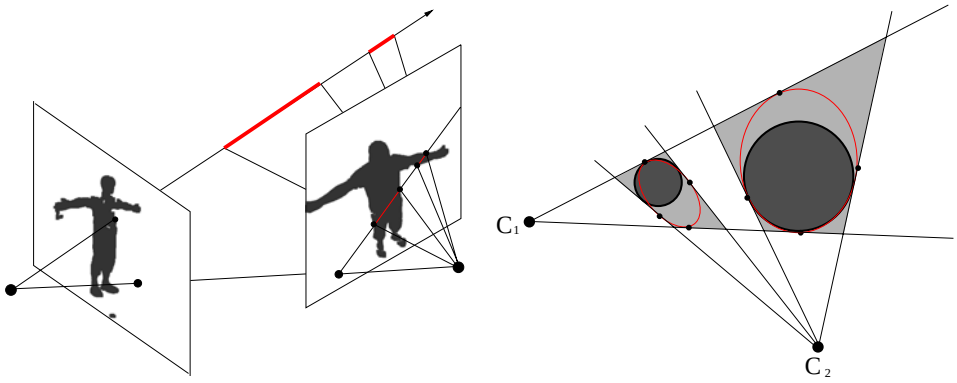

Figure 2: Left: two viewing edges along a viewing line computed solely from multiple-camera geometry. Right: two spheres (dark) that may be two distinct body parts, and the reconstructed surfaces (thin lines) with only two cameras. The *visual hull* is depicted by the shaded regions within the viewing lines originating in C_1 and C_2.

3-D points and surface normals. We explain now how to estimate the position and orientation of a *surface patch* that is supposed to lie onto the object's surface such that the latter is tangent to a viewing edge.

A viewing line associated with a silhouette from image j is tangent to the object's surface. If we assume that there is a unique viewing edge along a viewing line, then it means that this edge contains a surface point Y. Its orientation, a vector N, is defined by the cross-product between the viewing line and the tangent to the image silhouette. Notice that these computations can be carried out from image information only, provided that the calibration parameters of the camera are known.

The estimation of the position of point Y within a viewing edge requires some additional insights. Let Y belong to the viewing edge passing through the center of projection C_j of image j. This viewing edge is bounded by viewing lines associated with images i and k as well as their centers of projection C_i and C_k, as explained in the previous section. Since these viewing lines are tangent to the surface, we are also given these additional tangent directions – viewing lines originating in C_i and C_k – in the neighborhood of Y: The viewing lines from the silhouettes associated with images i and k which intersect the viewing line of Y . Under the assumption that the surface is locally of order 2, one can estimate the position of Y along a curve that lies onto the surface and which is constrained by three tangents. For farther details see [2].

The above reasoning applies to the case of a unique viewing edge along a viewing line. This is the case with most silhouette vertices if the cameras are evenly and sparsely distributed around the scene. However, this will not always be true, as shown on Figure 2. Whenever several viewing edges appear along the viewing line, the same approach is applied to each interval, one after one, thus producing as many 3D points and normals as the number of viewing edges. Note that not all the 3-D points thus determined actually belong to viewing edges tangent to the object's surface. Nevertheless, they all need to be computed in order to ensure that the local second order approximation of the surface is consistent with the visual hull. Moreover, as shown on Figure 2, the points thus obtained correspond to a better approximation of the object's surface

than the visual hull itself. This is particularly important when the task is to fit a curved model to the observations. Results obtained with this method are shown on Figure 3.

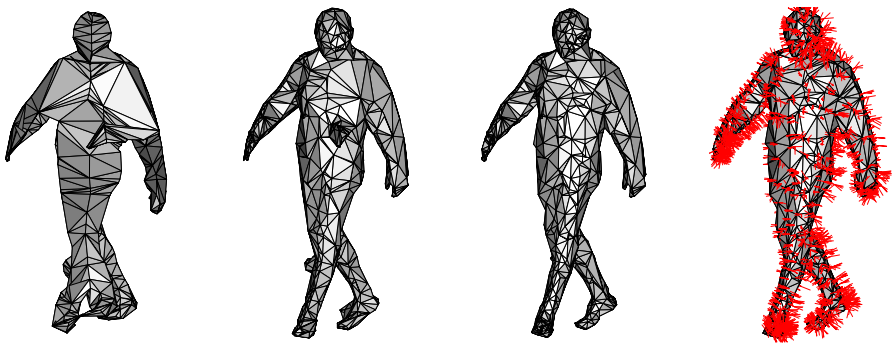

Figure 3: 3-D points (displayed as the vertices of a mesh) reconstructed with 2, 4, 5, and 6 cameras. The reconstructed normals are shown with the rightmost figure.

3 Modelling articulated objects

In order to model articulated objects such as human bodies, we use ellipsoids as basic volumetric shapes. These ellipsoids are joined and blended together to form an articulated implicit surface. In detail, an ellipsoid is a quadric described by a 4×4 homogeneous symmetric matrix $\mathbf{Q}$. This matrix is diagonal when the axes of the coordinate frame are aligned with the axes of inertia of the shape: $\mathbf{Q} = \mathrm{Diag}\,(1/a^2\, 1/b^2\, 1/c^2\, -1)$. The implicit equation of its surface writes $X^\top \mathbf{Q} X = 0$ where X describes the homogeneous coordinates of a 3-D point lying on this surface. The *signed algebraic distance* from a data point Y to this surface is $q(Y) = Y^\top \mathbf{Q} Y$. The value of q varies from -1 at the origin, to 0 on its surface, and then to $+\infty$ outside the ellipsoid as the point is farther away from the surface. It is convenient to use the exponential of the algebraic distance as a measurement error. The scalar parameter σ bounds the *distance of influence* of an ellipsoid, i.e.:

$$r(Y) = \exp\left(-\frac{q^2(Y)}{\sigma^2}\right) \tag{1}$$

When an ellipsoid undergoes a rigid motion, its matrix becomes $\mathbf{Q}_T = \mathbf{T}^{-\top} \mathbf{Q} \mathbf{T}^{-1}$ where $\mathbf{T}$ denotes a 4×4 homogeneous matrix associated with an Euclidean transformation. $\mathbf{T}$ describes a *free motion*, a *kinematic chain*, or a combination of both. In our case the articulated object has rotational joints with either one or three degrees of freedom. Such a mechanism may be described by a kinematic chain of the form: $\mathbf{T}_1 \ldots \mathbf{T}_k \ldots \mathbf{T}_n$ where each individual transformation is a one-parameter Lie group that can be decomposed into a fixed transformation followed by a rotation around an axis aligned with the mechanical axis (or with a virtual axis), and followed by the inverse of the fixed transformation, $\mathbf{T}_k = \mathbf{L}_k \mathbf{J}(\theta_k) \mathbf{L}_k^{-1}$. Matrix $\mathbf{T}_k$ describes the position and orientation of joint axis k with respect to a reference frame, and:

$$\mathbf{J}(\theta_k) = \begin{bmatrix} \cos\theta_k & -\sin\theta_k & 0 & 0 \\ \sin\theta_k & \cos\theta_k & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The fixed part of this transformation, $\mathbf{L}_k$, depends on the particular length of the kth joint and on the position and orientation of this joint with respect to a fixed reference frame. Within this paper we do not address the problem of estimating the exact size and shape of the object's joints and therefore this transformation will be provided.

We also consider the free motion of the object, a matrix $\mathbf{D}$. In the case of a human body in motion we attach the body frame to the torso and we make the simplification that the free motion of the torso is a 3-D translation. Therefore, matrix $\mathbf{D}$ can be parameterized by three translations along three orthogonal directions, $\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3$. Hence, the motion has n rotational joints and 3 free translations and is represented by $\Theta = (\theta_1 \dots \theta_n\ d_1\ d_2\ d_3)$; The motion transformation writes:

$$\mathbf{T}(\Theta - \Theta^0) = \mathbf{T}_1(\theta_1 - \theta_1^0)\dots\mathbf{T}_n(\theta_n - \theta_n^0)\mathbf{D}_1(d_1 - d_1^0)\mathbf{D}_2(d_2 - d_2^0)\mathbf{D}_3(d_3 - d_3^0) \tag{3}$$

This is known as the zero-reference representation of a kinematic chain because it describes the motion of each element of the object with respect to a fixed reference pose Θ^0 that can arbitrarily be chosen [13]. In the case of tracking, we seek the pose of the object at a time instant t *provided that the pose at the previous time instant* $t-1$ *has been already determined*, and therefore we can choose the pose of the object (and hence the pose of each one of its elements) associated with the previous time instant as the zero-reference pose: $\mathbf{T} = \mathbf{T}(\Theta^t - \Theta^{t-1})$. The matrix of an ellipsoid at time t can now be expressed as a function of the motion parameters, i.e., $\mathbf{Q}(\Theta^t) = \mathbf{T}^{-\top}\mathbf{Q}(\Theta^{t-1})\mathbf{T}^{-1}$.

We consider a complete object model. In particular a human body model with 22 rotational degrees of freedom is a relatively complete model that allows to capture the most general human actions. Therefore, there is a total of $22+3$ degrees of freedom, i.e., Θ is of dimension 25. Moreover, body parts are described by ellipsoids denoted by $\mathbf{Q}_1$, $\mathbf{Q}_2$, and so forth. Obviously, there is a kinematic chain for each body part and the number of degrees of freedom are different for each one of these chains. There is a quadratic form or a signed algebraic distance $q_i(X)$ associated with an ellipsoid $\mathbf{Q}_i$ as well as an exponential algebraic distance r_i, i.e., eq. (1); For an object in motion we have $q_i(X,\Theta)$ and $r_i(X,\Theta)$.

An articulated implicit surface can now be defined at each time instant as a level-set of a blending of these ellipsoids [15]:

$$f(X,\Theta) = \sum_{i=1}^{22} r_i(X,\Theta) = 1 \tag{4}$$

4 Fitting and tracking

It is now possible to formulate the problem of tracking an articulated shape as the problem of fitting the model to the data [6]. At each time instant the following minimization problem has to be solved:

$$\min_{\Theta} F(\Theta) = \left(\sum_{j=1}^{m} \beta_j (f(Y_j,\Theta) - 1)^2 \right) \tag{5}$$

where the weight β_j describes the probability of a data point Y_j to be consistent with the model, $\beta_j = \exp(-(f(Y_j,\Theta)-1)^2/\sigma^2)$. A large value for σ allows virtually all the data points to contribute to the fit, including data points that are far away from the model. A smaller value for σ allows to limit the influence of a datum to nearby quadrics. Within an Expectation-Maximization

formulation such as in [6] an iterative procedure decreases the value of σ as the fitting proceeds. This allows, in principle, to escape from local minima when there is a large discrepancy between the data and the model pose. It also allows to disregard outliers at the final iterative steps of the algorithm. Surface orientation information is not taken explicitly into account. Ellipsoids whose local surface normals are very different will equally contribute when associated with a datum. We will modify the error function of eq. (5) in order to explicitly take into account *surface normals*.

The scaled algebraic distance. One important merit of any visual tracking method is its speed. Eventually tracking should be implemented in real time, i.e., compatible with the frame rates delivered by the cameras. Therefore, there is a compromise to be made between complexity and efficiency. The computation of the distance between an observation and the model resides in the inner loop of the tracker, and therefore it must be efficiently computed. The algebraic distance is fast to compute but has drawbacks. The Euclidean and pseudo-Euclidean distances are more expensive [6].

Let $\mathbf{Q}$ be an ellipsoid with parameters a, b, and c. Notice that matrices $\mathbf{Q}$ and $\lambda\mathbf{Q}$, with $\lambda \neq 0$ describe the same quadric. However the algebraic distances to these ellipsoids are different. Let $r^2 = a^2 + b^2 + c^2$. The scaled algebraic distance from a point Y to the ellipsoid is defined by $q^r(Y) = r^2 Y^\top \mathbf{Q} Y$. When the ellipsoid is close to a sphere and when the observation is close to its surface, the scaled algebraic distance is a good approximation of the Euclidean distance. However, with substantially elongated ellipsoids, the scaled algebraic distance does not introduce any improvements. Such an effect is known as high curvature bias. The practical solution that may be easily adopted consists in replacing elongated ellipsoids by an equivalent number of spheres.

Using surface orientation constraints. So far we used data points and we did not take into consideration the normals available with the 3-D observations. Let $N = (n_1\ n_2\ n_3\ 0)^\top$ be the vector normal to the surface patch and let $[N]_3$ denote the 3-vector formed with n_1, n_2, n_3. We also have $N^\top N = 1$.

It is well known that the 4-vector $P = \mathbf{Q}X$ defines the equation of a plane P tangent to the quadric at point X lying on its surface [11]. Therefore the 3-vector $[P]_3$ designates the normal vector to that plane. When a surface patch is consistent with the model, vectors $[P]_3$ and $[N]_3$ are aligned, therefore their cross-product is null and their dot-product is equal to either $+1$ or -1. A measurement of the discrepancy between a surface patch orientation and the nearby model orientation must use the followings:

$$d(Y,N,\mathbf{Q}) = [N]_3 \times [\mathbf{Q}Y]_3 \quad , \quad \alpha(Y,N) = \frac{1}{2}\left(1 - \frac{N^\top \mathbf{Q} Y}{\|\mathbf{Q}Y\|^2}\right)$$

The first one of these measurements, d, is equal to zero for a perfect match but is defined up to a 180^0 ambiguity. The second measurement, α varies between 0 (for vectors with opposite orientation) and 1; Therefore it may act as a normalized measure of a plausibility.

As in the case of point data, we define the exponential distance from an observation (a 3-D point Y and a normal N) to the 22 ellipsoids forming the model:

$$g(N,Y,\Theta) = \sum_{i=1}^{22}\left(\alpha_i \exp\left(\frac{-d(Y,N,\mathbf{Q}_i(\Theta))}{\mu}\right)\right) \tag{6}$$

Hence, one obtains an optimal solution by fitting all the 3-D observations to the model:

$$\min_{\Theta} G(\Theta) = \left(\sum_{j=1}^{m} (g(N_j, Y_j, \Theta) - 1)^2 \right) \tag{7}$$

Tracking articulated objects. In order to track articulated objects we minimize a linear combination of the error functions $F(\Theta)$ and $G(\Theta)$; The first one of these functions, eq. (5), fits the locations of the observations with the model while the second one, eq. (7) fits the normals of the observations with the same model:

$$\min_{\Theta}(\omega_1 F(\Theta) + \omega_2 G(\Theta)) \tag{8}$$

The tracking does not need segmentation of the data. Observations at time t are handled totally independently of observations at time $t-1$. The solution previously found, Θ^{t-1} is used in conjunction with a Kalman filter, and with a constant angular velocity hypothesis, in order to initialize the tracker at time t. Joint limits were set and added as penalty terms to the objective function in order to prevent unnatural human postures.

Another issue is the choice of ω_1 and ω_2 in eq. (8). These weights balance the contribution of position and orientation. There are methods allowing to initialize these weights and to modify them during the minimization process. However, as explained in the next section, we found that there are many advantages in using both position and orientation constraints. Therefore we chose $\omega_1 = \omega_2 = 1$.

5 Experiments

We validated the method with both simulated and real data. The former was obtained using a human animation software package. The latter was obtained with 6 calibrated cameras. Sequences of image silhouettes were generated with the animation software. Then the method described above was applied to these data. The simulated data allowed us to (i) assess the quality of the tracker with respect to a ground truth, (ii) analyse the behavior in the presence of Gaussian noise added to the data, (iii) quantify the merit of using surface normals, and (iv) determine the optimal number of observations needed to reliably estimate an object pose. Figure 4 illustrates some of the results out of a large number of experiments. From performing all these experiments one may conclude that tracking is notoriously improved when surface patches are used rather than just points. The surface-patch based objective function, i.e., eq. (8) converges faster, allows for less 3-D observations, and is more tolerant to errors in position. Figure 5 shows the results of applying the method to a 4 second sequence (120 frames) and with six cameras.

6 Discussion and conclusions

In this paper we described a method for tracking the motion of articulated objects. At each time instant, the images are segmented into foreground and background thus providing a set of 2-D silhouettes. These silhouettes are combined together with multiple-camera geometric constraints and with a simple assumption about the surface of the object in order to estimate 3-D surface patches: points and normals. The model itself is an *articulated implicit surface* combining a zero-reference kinematic chain with a set of ellipsoids. The model is fitted to the 3-D observation by minimization of an objective function that takes into account both the location

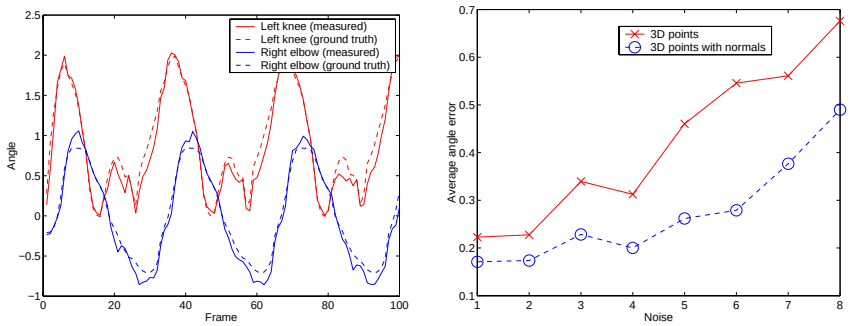

Figure 4: Left: comparison between the true angles and the estimated joint angles: left knee (top) and right elbow (bottom). Right: sensitivity to noise for points only (top) and for points and normals (bottom). The method always performs better when normals are taken into account. The two curves show the average angular error as a function of noise.

of these observations and their 3-D orientations. The resulting tracker is very efficient, it can deal with noisy data and with outliers, and it does not require data-to-object-part assignments.

Interesting enough, augmenting the number of cameras increases the robustness of the method without affecting its efficiency, since an increased number of cameras provides more precisely located surface patches. In practice we think that the optimal number of cameras is between 4 and 6.

Certainly, there are methods able to recover articulated motion with a single camera. These methods need sophisticated probabilistic methods to work well. They require a learning phase. We believe that our method is a potential candidate for providing data needed by learning methods.

In the future we plan to build a complete bio-mechanical model of humans with 80 degrees of freedom. We also plan to relax some of the constraints currently limiting our method, such as the requirement to have relatively accurate closed 2-D silhouettes. Finally, based on our fitting method, we plan to implement the bootstrapping of the tracker using a coarse-to-fine representation of the joint space and a hierarchical description of an articulated object.

References

[1] A. Agarwal and B. Triggs. Learning to track 3D human motion from silhouettes. In *International Conference on Machine Learning*, pages 9–16, Banff, July 2004.

[2] E. Boyer and M.-O. Berger. 3D surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997.

[3] E. Boyer and J.-S. Franco. A hybrid approach for computing visual hulls of complex objects. In *Computer Vision and Pattern Recognition*, pages 695–701, June 2003. Madison, Wisconsin, USA.

[4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara CA, pages 8–15, 1998.

[5] D. Demirdjian and T. Darrell. 3-D articulated pose tracking for untethered diectic reference. In *Proceedings of ICMI'02*, Pittsburgh, Penn., October 2002.

Figure 5: Top: Frames 17, 46, 59, and 81 for one of the six cameras. Bottom: The corresponding pose of the articulated implicit surface shown with the same camera parameters as above.

[6] G. Dewaele, F. Devernay, and R. Horaud. Hand motion from 3d point trajectories and a smooth surface model. In T. Pajdla and J. Matas, editors, *8th European Conference on Computer Vision*, volume I, *LNCS 3021*, pages 495–507. Springer, May 2004.

[7] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *Proceedings of the Eighth International Conference on Computer Vision*, volume II, pages 315–320, Vancouver, Canada, July 2001.

[8] N. R. Howe, M. E. Leventon, and W. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Advances in Neural Information Processing Systems*, Denver, volume 12, pages 820–826, 1999.

[9] I. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000.

[10] G. Cheung, T. Kanade, J.Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 714 – 720, June 2000.

[11] Q.-T. Luong and O. D. Faugeras. *The Geometry of Multiple Images*. MIT Press, Boston, 2001.

[12] W. Matusik, C. Buehler, and L. McMillan. Polyhedral Visual Hulls for Real-Time Rendering. In *Eurographics Workshop on Rendering*, 2001.

[13] J. M. McCarthy. *Introduction to Theoretical Kinematics*. MIT Press, Cambridge, 1990.

[14] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[15] R. Plänkers and P. Fua. Articulated Soft Objects for Multi-View Shape and Motion Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1182–1187, 2003.