Automatic Determination of Shoulder Joint Limits using Quaternion Field Boundaries

Lorna Herda, Raquel Urtasun, Pascal Fua*
Computer Vision Laboratory
EPFL
CH-1015 Lausanne, Switzerland
{Lorna.Herda, Raquel.Urtasun, Pascal.Fua@epfl.ch}

Andrew Hanson
Computer Science Department
Indiana University
Bloomington, IN 47405 USA
hanson@cs.indiana.edu

International Journal of Robotics Research, 22(6), June 2003

Abstract

To provide efficient tools for the capture and modeling of acceptable virtual human poses, we propose a method for constraining the underlying joint structures based on real life data. Current tools for delimiting valid postures often employ techniques that do not represent joint limits in an intuitively satisfying manner, and furthermore are seldom directly derived from experimental data.

Here, we propose a semi-automatic scheme for determining ball-and-socket joint limits by actual measurement and apply it to modeling the shoulder complex, which—along with the hip complex—can be approximated by a 3 degree-of-freedom ball-and-socket joint. Our first step is to measure the joint motion range using optical motion capture. We next convert the recorded values to joint poses encoded using a coherent quaternion field representation of the joint orientation space. Finally, we obtain a closed, continuous implicit surface approximation for the quaternion orientation-space boundary whose interior represents the complete space of valid orientations, enabling us to project invalid postures to the closest valid ones.

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

1 Introduction

In the world of animated virtual humans, with flawless skin textures and fluid motion, we must give credit to the animators who have driven the illusion to the state of near perfection that it has now reached. Every ripple on the skin is painstakingly refined until it no longer shocks our eye or disappoints our expectations. Yet, for all the available animation tools, it is still up to the animator to manually constrain postures to valid ranges. Similarly, in the computer vision world, many approaches to tracking and modeling people from video sequences have been proposed [Gavrila, 1999, Moeslund and Granum, 2001]. They are not always robust, in part because image data are inherently noisy and in part because it is inherently ambiguous [Rehg *et al.*, 2003]. Introducing valid joint limits is therefore one important practical step towards restricting motion synthesis and analysis algorithms to humanly feasible configurations, thereby increasing their reliability and potential for automation.

Current tools for delimiting valid postures often employ techniques that do not represent joint limits in an intuitively satisfying manner. They tend to be expressed in terms of hard limits on individual rotation angles that do not account for dependencies between those angles [Tolani *et al.*, 2000, Wang *et al.*, 1998], such as those between the allowable amount of arm twisting and the arm's position.

By contrast, in this work, we propose a quaternion-based model that allows us to explicitly represent and measure those dependencies: We first measure the joint motion range using optical motion capture. We next convert the recorded values to joint poses encoded by a coherent quaternion field representation of the joint orientation space. Finally, we derive a closed, continuous implicit surface approximation for the quaternion orientation-space boundary whose interior represents the complete space of valid orientations, ultimately enabling us to project any invalid posture to the closest valid one.

Quaternions represented as points in a three-sphere [Shoemake, 1985], the axis-angle representation, and the corresponding exponential maps may all be used to specify an orientation frame in such a way that the Gimbal Lock singularities of traditional Euler angles are avoided. However, only quaternions are endowed with an intrinsic natural distance between orientation frames. We choose the quaternion representation in this work because of this fact: Quaternions provide the most robust computational framework for enforcing joint-angle constraints by projection onto the nearest point in the subspace of valid orientations.

The dependency between arm twist and arm directional orientation, or swing, is a function of the complicated make-up of the shoulder complex, which we were given the opportunity to study extensively in the context of research projects carried out in our laboratory. We therefore chose this example to validate our approach to expressing ball-and-socket joint limits. Because similar constraints exist for hip and leg motion, the hip joint would have been another natural candidate. Modeling the shoulder complex as a ball-and-socket joint, one with exactly three independent degrees of rotational freedom and no more, is of course an oversimplification. In reality, it is composed of three joints, the gleno-humeral, scapula, and clavicle. However, for applications such as video-based motion tracking, these three joints are almost unobservable and the only thing that can be measured is the position of the arm with respect to the body. The joint limit representation we propose can then be understood as a way of encoding the space of reachable arm positions without having to precisely describe the shoulder's anatomy.

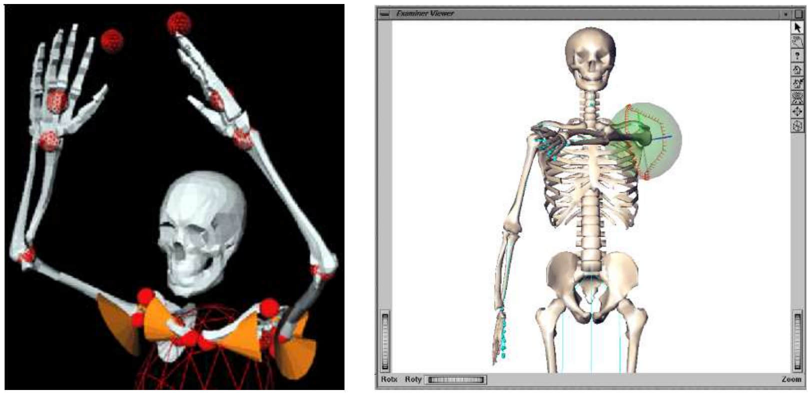
In the remainder of the paper, we first briefly review the state of the art. We then introduce our approach to experimentally sampling the space of valid postures that the joint allows and to representing this space in terms of an implicit surface in quaternion space. Finally, we demonstrate our method's effectiveness for animation purposes and discuss its applicability in a tracking context.

2 Related approaches

In the field of biomechanics, joint limits have often been measured, using various mechanical and electromagnetic devices. Examples of such statistical data can be found in [Johnston and Smidt, 1969] for the hip, in [Engin and Tumer, 1993] for the knee, and in [Meskers *et al.*, 1998] for the shoulder. Previous studies, such as [Wang *et al.*, 1998], have shown inter-subject variance to be extremely small. Thus, it is acceptable to generalize results obtained on the basis of measurements carried out on a standard individual without motion impairment, or on a very small number of subjects.

Many of these empirical results are subsequently re-used for modeling human skeletons, when the conversion from the measured data to the representation is readily feasible. The simplest approach is to introduce joint hierarchies formed by independent 1-Degree-Of-Freedom (DOF). These DOFs are often described in terms of Euler angles and the joint limits given as minimal and maximal values. For tracking purposes, these angles or their evolution from frame to frame can then be linearized [Bregler and Malik, 1998], which simplifies the computation and results in increased speed. This formalism, however, does not account for the coupling of the different limits and, as a result, does not properly account for the 3-D accessibility space of real joints. Euler angles suffer from an additional weakness known as "Gimbal lock." This refers to the loss of one rotational degree of freedom that occurs when a series of rotations at 90 degrees is performed, resulting in the alignment of the axes [Bobick, 1998, Watt and Watt, 1992]. Sinus cones [Engin and Tumer, 1989] improve upon this situation and are one of the most widely-used representations. This scheme was later adopted for 3-D bio-mechanical modeling of the human upper-limb joints [Maurel, 1998]. In this work, joint boundaries are represented by conic shapes with elliptic bases such as those shown in Fig. 1(a).

The spherical polygons [Korein, 1985] depicted by Fig. 1(b) form the basis for another popular formalism to express joint limits, which has been extensively used in the areas of animation and inverse kinematics [Baerlocher and Boulic, 2000]. Recently, a new model in which the workspace of angular motion is modeled as a triangular Bezier spline surface has been proposed [Tolani *et al.*, 2000]. Measurements were performed and the method was validated by comparison with statistical data available in the literature. The 3 DOFs of a ball-and-socket joint are decomposed into an angular and an axial component. The angular direction component, or swing, corresponds to the two flexion and abduction DOFs, whereas the ax-



(a) (b)

Figure 1: Joint limits representation. (a) Joint sinus cones. (b) Spherical polygon.

ial component, or twist, is the rotation around the axis of the upper arm limb [Korein, 1985, Grassia, 1998].

In all the cases discussed above, limits for swing and twist are handled independently, even though, in reality, they are correlated. This correlation was measured by recording arm motions and expressed in terms of the amount of allowable twist as a function of swing in [Wang *et al.*, 1998]. This work, however, did not propose a model that simultaneously provides joint limits for all the DOFs.

In short, the method we propose here advances the state-of-the-art because it provides a way to enforce limits on swing and twist while at the same time accounting for their dependencies. Furthermore, the quaternion representation we use is not subject to singularities such as the “Gimbal lock” of Euler angles or mapping rotations of 2π to zero rotation. Finally, the well-defined distance measure between orientation defined by quaternions supports uniquely appropriate methods for finding the nearest physically correct orientation corresponding to an erroneous initial joint position. Although other representations could have been used to achieve these goals, the unit quaternion, which denotes each orientation as one of a pair of mirror-image points on the three-sphere [Shoemake, 1985], has several unique properties: The existence of a valid distance measuring the proximity of two orientations, the ability to use that measure to uniquely determine the nearest physically correct orientation corresponding to an erroneous initial one, and the ability to interpolate with meaningful angular velocity constraints on an optimal path splining among orientation key frames. Therefore, even though the quaternion formalism introduces an additional parameter and constraint [Grassia, 1998], we have chosen it as the most logical basis for this work.

3 Measuring and Representing Shoulder Motion

We used the Vicon^{tm} motion capture system to collect 3-D marker data corresponding to extensive shoulder movements along the boundaries. The set of possible joint orientations and positions in space can be considered as a path of referential frames in 3-D space [Bloomenthal, 1990], which can then be mapped to a sub-space of the space of possible quaternion referential frames of the object, that we call the quaternion Gauss map [Hanson, 1998b]. In other words, paths in the space of possible referential frames correspond to quaternion referential frames on this map.

In practice, we represent rotations by the sub-space of unit quaternions S^3 forming a unit sphere in 4-dimensional space. Any rotation can be associated to a unit quaternion but we need to keep in mind that the unitary condition needs to be ensured at all times. A rotation of θ radians around the unit axis *v* is described by the quaternion:

$$q = [q_x, q_y, q_z, q_w]^T = [sin(\frac{1}{2}\theta)v, cos(\frac{1}{2}\theta)]^T \tag{1}$$

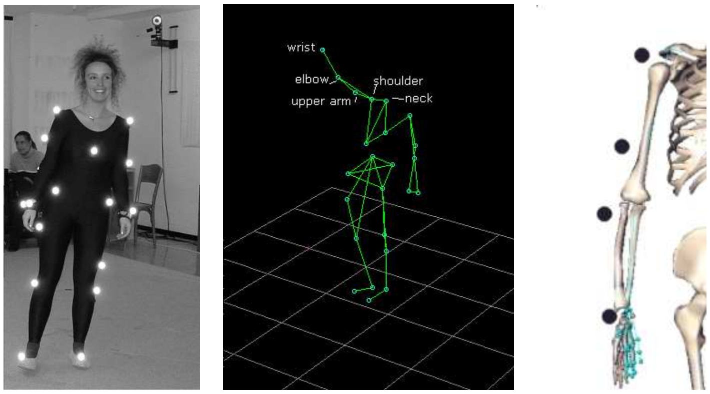
Since we are dealing with unit quaternions, the fourth quaternion component q_w is a dependent variable and can be deduced, up to a sign, from the first three. Given data collected using optical markers, we obtain a cloud of 3-D points by keeping the spatial or (q_x, q_y, q_z) coordinates of the quaternion. In other words, these three numbers serve as the coordinates of quaternions expressed as projections on three conventional Cartesian axes.

Because we simultaneously measure swing and twist components, and because the quaternion formalism lets us express both within one rotation, this representation captures the dependencies between swing and twist, one of the major goals of this work. In this manner, we will have generated joint limits on the basis of motion capture. We will then be able to make use of these joint limits as constraints for tracking and pose estimation, by eliminating all invalid configurations.

3.1 Motion measurement

We captured shoulder motions using the Vicon Motion Capture System, with 19 strategically-placed markers to recover the whole body motion of an actor. Figure 2 shows the actor wearing the complete set of markers, enabling the system, if needed, to identify the entire set of markers, not only those around the shoulder. We used a complete standard marker configuration, given that hierarchical connections and segment lengths are used for marker tracking and identifying. Capturing the markers only on the upper arm would lead to a higher level of ambiguity, thereby causing potential marker labeling confusion.

For our purposes, we normally take into consideration only the four markers depicted by Fig. 2(c) and track their motion over the whole captured sequences. These markers are those placed on the shoulder joint, on the upper arm segment, at the elbow and on the wrist. An additional marker is placed at neck level to serve as a fixed reference. As the subject moves in 3-D space, the Vicon system reconstructs the markers in 3-D global space for every image frame of the sequence, and then labels those that it is able to identify. This enables us to retrieve the global 3-D positions of our five markers over the entire sequence. This is sufficient to retrieve the upper arm orientation, or in other words the motion range of the shoulder joint. To generate a dense data set that covers the entire range of possible shoulder motions, we



(a) (b) (c)

Figure 2: Vicon Motion Capture session, (a) Actor with markers. (b) Reconstructed 3-D markers. (c) Relative position of the markers with respect to the skeleton.

captured motion data for several minutes during which the subject attempted to perform as complete as possible a set of motions such as those depicted by Fig. 3.

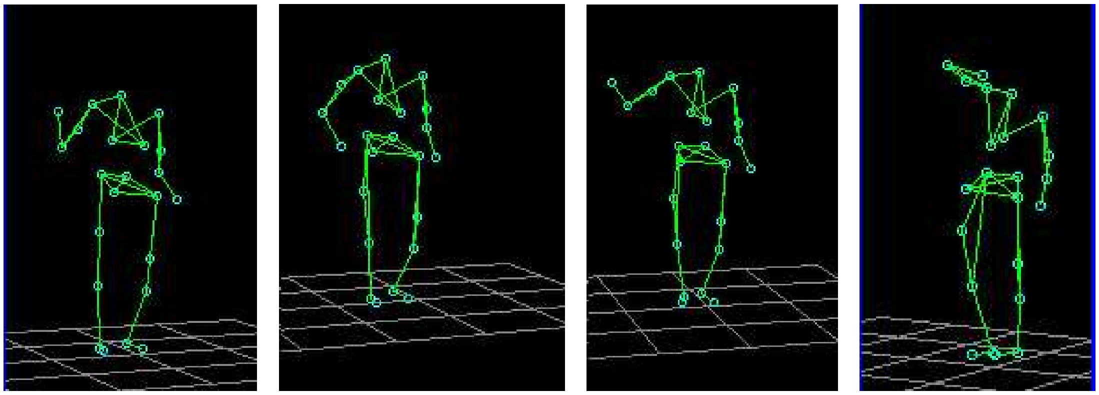


Figure 3: Shoulder motion sequence.

3.2 Motion representation

Quaternion field technology

Our goal is to create an analyzable topographic space of joint orientations using the experimental 3-D marker coordinates as the basis. To accomplish this, we adopt the method of quaternion fields described in[Hanson, 1998a, Hanson and Ma, 1995]. In the quaternion field method, each joint orientation is first converted to a 3×3 matrix M, which, using Euler's theorem[1], can be expressed in terms of its lone real eigenvector $\hat{\mathbf{n}}$ and the angle of rotation θ

[1] Not to be confused with Euler angles!

about that axis. This in turn may be expressed as a point in quaternion space, or, equivalently, a point on a three-sphere S^3 embedded in a Euclidean 4D space. To find θ and axis $\hat{\mathbf{n}}$, given *any* rotation matrix or frame M, we need two steps:

1. Solve for θ using
$$\mathrm{Tr}M = 1 + 2\cos\theta \ ; \tag{2}$$
2. Find $\hat{\mathbf{n}}$ from
$$M - M^t = \begin{bmatrix} 0 & -2n_3\sin\theta & +2n_2\sin\theta \\ +2n_3\sin\theta & 0 & -2n_1\sin\theta \\ -2n_2\sin\theta & +2n_1\sin\theta & 0 \end{bmatrix} \tag{3}$$
which is defined as long as $\theta \neq 0$. In the degenerate case, the matrix and the quaternion are taken to be the identity transformation. For more details, see Biedenharn and Louck [Biedenharn and Louck, 1981].

The identification of the corresponding quaternion follows immediately from

$$q(\theta, \hat{\mathbf{n}}) = (\cos\frac{\theta}{2}, \hat{\mathbf{n}}\sin\frac{\theta}{2}) \tag{4}$$

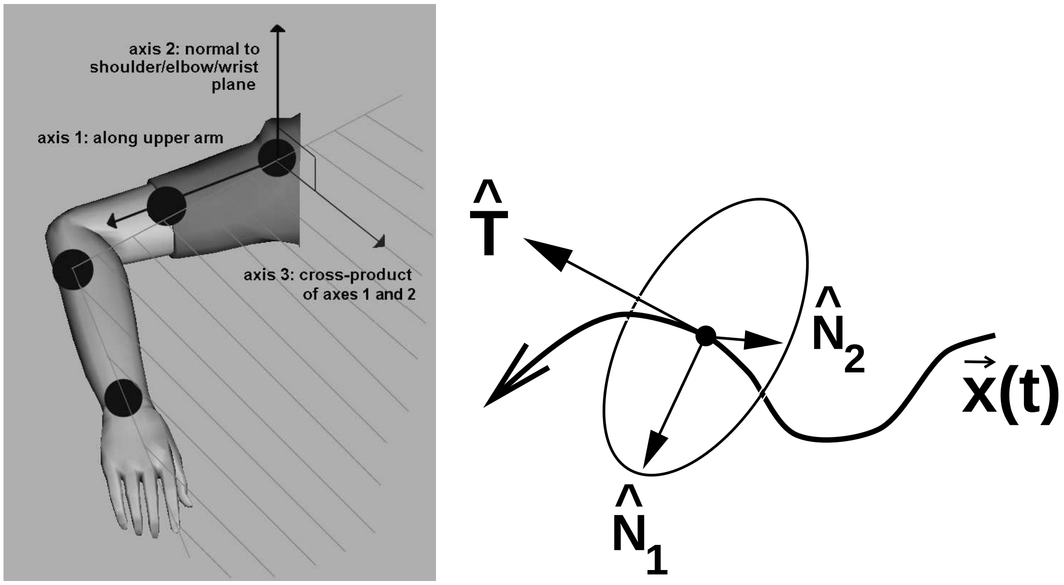
up to the sign ambiguity between the two equivalent quaternions q or $-q$, which correspond to the exact same rotation matrix M. A single compact volume containing a dense set of quaternion frames will have two distinct but equivalent realizations due to the $q \Leftrightarrow -q$ ambiguity. We have implemented various procedures to cast out duplicates, or, equivalently, to iteratively check the 4D dot-product between neighboring quaternions and force all neighbors to have signs that are consistent with positive dot products.

Construction of shoulder coordinate frames

To build our frames from the marker coordinates, we begin by subtracting the neck marker coordinates from the other four coordinates, so that all coordinates are expressed with respect to a fixed point.

The frames are then constructed as follows: The first axis of the frame corresponds to the upper arm segment, i.e. the line defined by the shoulder and upper arm markers. The second axis is the normal to the plane passing through the shoulder, elbow and wrist markers. This plane represents the axial rotation, or “twist” motion, of the shoulder joint. The third axis simply is the axis orthogonal to the other two, that is their cross-product. The construction of the referential is illustrated by Fig. 4(a).

An orthonormal coordinate frame is created in this manner for each image frame of the sequence, along with the rotation relating one image frame to the next. The calculated 3 × 3 rotation matrices $M(t)$ for each data point are then converted into unit quaternions as described above, producing a dense cloud of points occupying a volumetric portion of the quaternion 3-manifold or three-sphere.


axis 2: normal to shoulder/elbow/wrist plane
axis 1: along upper arm
axis 3: cross-product of axes 1 and 2
$\hat{T}$
$\hat{N}_2$
$\vec{x}(t)$
$\hat{N}_1$

(a) (b)

Figure 4: (a) Deriving a referential from the markers. (b) General form of a moving frame for a 3-D curve $\mathbf{x}(t)$, with the tangent direction $\hat{\mathbf{T}}$ determined directly from the curve derivative, and the exact orientation of the basis $(\hat{\mathbf{N}}_1,\ \hat{\mathbf{N}}_2)$ for the normal plane determined only up to an axial rotation about $\hat{\mathbf{T}}$

The concept of a generic moving frame on a curve is illustrated by Figure 4(b). When transformed to quaternion form, each successive frame from the moving sequence becomes a point on a quaternion curve lying within the three-sphere S^3 embedded in R^4. Families of frames on a surface become surfaces in quaternion space, and collections or disjoint sets of orientations such as those collected here for the shoulder joint become a cloud of points occupying a volumetric quaternion space with a distinct closed boundary surface.

By retaining only the spatial or (q_x, q_y, q_z) coordinates of the quaternion, we can represent the four motions of Fig. 5 as trajectories in 3–D space. By processing in this fashion the long motion capture sequence discussed in Section 3.1, we obtain the cloud of 3–D points depicted by Fig. 6(a).

4 Volumetric Approximation of Scattered 3-D Data

To make effective use of the collected data, we now need to derive a working representation for it. In the 3-D quaternion sub-space *(q_x, q_y, q_z)*, we chose to approximate the cloud of points representing all the valid measured shoulder rotations by an implicit surface. It will enable us to decide whether a movement is possible or not, using a simple inside/outside test.

Many other methods have been studied to solve the shape reconstruction problem [Bittar *et al.*, 1995]. We preferred the implicit surface representation because it allows us to express the

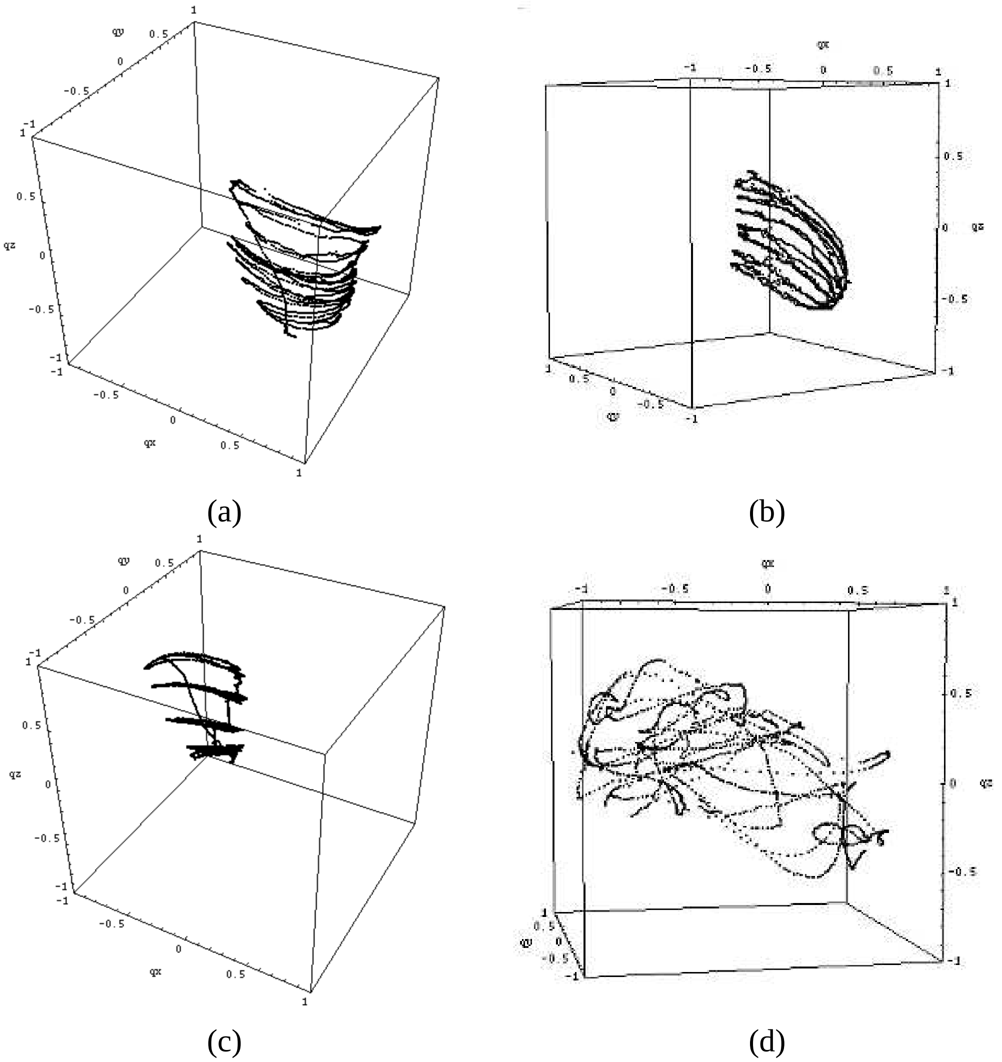
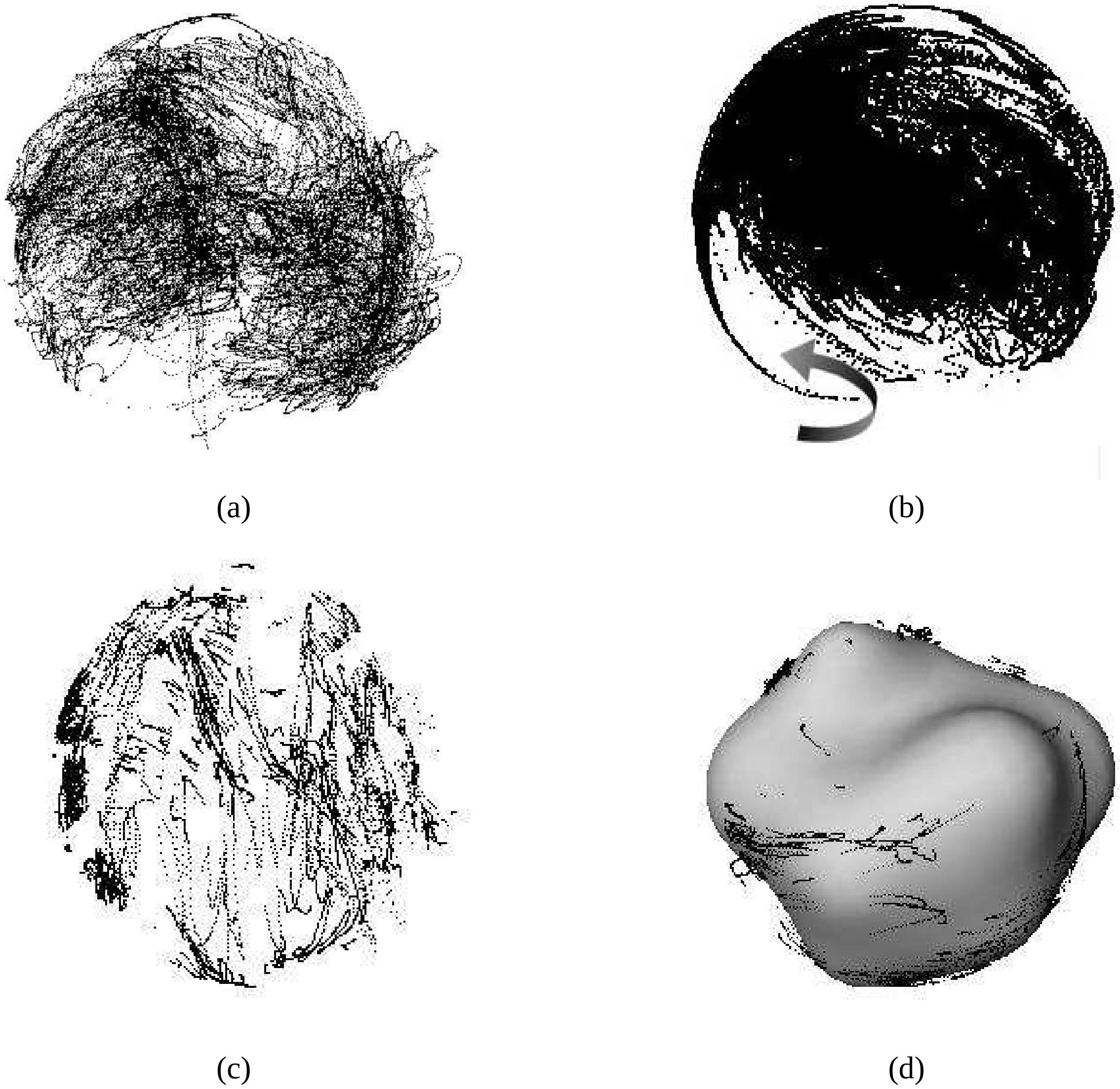


Figure 5: Decomposing the motion. (a) Flexion. (b) Abduction. (c) Twist at four different positions. (d) Random upper arm motions.

shape using relatively few parameters, while at the same time providing a convenient metric for estimating proximity. This is due to the fact that the surface is represented by an equation and not modeled explicitly [Opalach and Maddock, 1995]. Not only will an implicit surface provide us with a smooth representation of the volume of valid quaternion rotations, but thanks to its well-defined distance function between data points and the obtained surface, the fitting process becomes relatively simple, as does the final inside/outside decision. Furthermore, when dealing with a value representing an invalid rotation, the use of implicit surfaces gives us the ability to easily project this value to the closest valid one. This will be further discussed in Section 5.

To the best of our knowledge, in our field, the idea of using implicit surfaces for the purpose of approximating scattered data was first reported in[Muraki, 1991]. This work introduced the


(a)

(b)

(c)

(d)

Figure 6: From the 3-D data points to the fitted implicit surface. (a) Original volume of 3-D quaternion data points. (b) Visible gap within the data. (c) Extracted surface points. (d) Final fitted implicit surface representing the shoulder joint limits.

notion of implicit iso-surfaces generated by shape primitives. These primitives are defined by their skeletons that can either be points—the center of a sphere or ellipsoid—or extend features—a line or a polygon. The algorithm positions a skeleton at the center of mass of the points and subdivides it until the process reaches a correct approximation level. This, however, has several drawbacks, such as the lack of a local criterion and a high computational cost. Furthermore, the algorithm does not perform well on data with "holes" and requires knowledge of surface normals.

This early work was later extended [Bittar *et al.*, 1995, Tsingos *et al.*, 1995] by introducing a more efficient way to split the implicit surfaces without requiring normals and expressing them in terms of locally defined field functions. This allows the use of an iterative method based on minimizing the distance between the real points and the generated surface. This method forms the basis of our approach, but has required substantial modifications because:

- As can be seen in Fig. 6, our motion capture data form a volume in quaternion space, as opposed to a surface and, therefore, the surface points must first be recovered;
- Despite our best efforts, the complete range of shoulder motions was not equally covered, resulting in variations in data density and even "holes," such as the one highlighted in 6(b), where no data are available;
- The original method required the user to position some initial surface skeletons inside the volume to reconstruct and automation was achieved using the medial axis transform, which is impractical in our case because our data are far too noisy.

These issues are discussed in detail below.

4.1 From Volume to Surface

Before performing any implicit surface approximation, we need to address the fact that our point cloud forms a 3-D volume instead of a 3-D surface. To transform the points into a surface, we apply a method based on dividing the 3-D space into overlapping solid angles placed at the center of mass of the cloud. To this end, the 3-D Cartesian points $X_j(x, y, z)$ are converted into spherical coordinates $X_j(\rho, \theta, \phi)$. The 3-D space is then divided into solid angles by taking into account the points that fall within the intervals $[\theta, \theta + d\theta]$, $[\phi, \phi + d\phi]$.

Let $SP(i, k)$ be the points inside the (i, j) solid angle and let $\rho_{max}(i, j)$ be their maximum radius:

$$\rho_{max}(i, k) = \max_{x_j \in SP(i,k)} \rho_j \quad .$$

The surface points are taken to be those whose radius is close to ρ. For each solid angle we write

$$Surface(SP(i, k)) = \{x_j(\rho_j, \theta_j, \phi_j) \in SP(i, k), (1 - T)\rho_{max}(i, k) < \rho_j \leq \rho_{max}(i, k)\} \quad ,$$

where T is an appropriate threshold value. The complete surface becomes the union of all the surface points

$$Surface(SP) = \bigcup_{i=1}^{L} \bigcup_{k=1}^{L} Surface(SP(i, k)) \quad ,$$

with L being the number of quantized intervals for the θ and ϕ angular coordinates. Some histograms, representing the point density as a function of the radius ρ are shown in Fig. 7. In Fig. 8 we can see the projected data volume for one particular solid angle of the actual shoulder data and the resulting surface points.

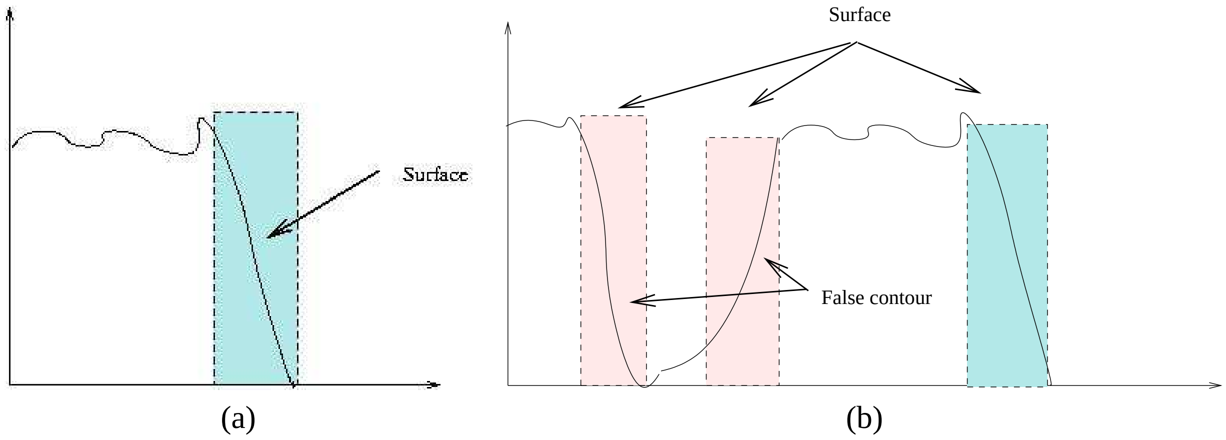


Figure 7: (a) Histogram of the point density as function of the distance for a solid angle containing dense data. The surface points are taken to be those in the box. (b) Histogram of a solid angle containing one hole. The contours of the hole, that is the points inside the two boxes at the left, are ignored and only the outer surface is detected.

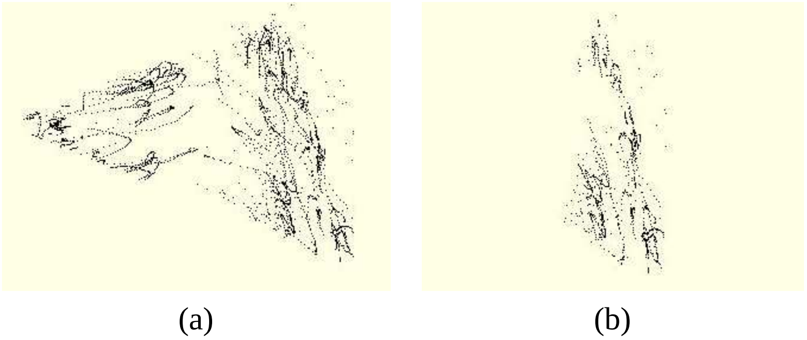


Figure 8: Surface extraction using real data. (a) Data points in one of 49 solid angles used to cover the space of orientations. (b) Corresponding surface points, extracted using a distance threshold $T = 0.1$.

The strength of the process is that it does not require the entire space to be densely populated by data points. As discussed above, this is important because the complete range of shoulder motion could not be equally covered, resulting in variations in data density. Its weakness is that it depends on two arbitrary values, L that controls the number of solid angles and T that defines the "thickness" of the surface boundary. The higher the curvature and greater the density of data points, the finer the quantization and the smaller the thickness ought to be. However, in the presence of noise or of relatively sparse data, both the size of the solid angles and the boundary thickness should be large enough to yield a relatively smooth surface without holes. In practice, the choice of the L and T parameters are therefore heuristic and problem dependent.

For our specific shoulder modeling problem, we have found that using $L = 7$ and $T = 0.1$ yields acceptable results. This yields the surface points depicted by Fig. 6(c). A more generic solution would be to develop a marching-cube based method that divides the space into voxels and finds the boundary between those that contain high and low data point densities.

4.2 Implicit Surface Parameterization

In our implementation, we take the implicit surface $\mathcal{S}$ to be an iso-surface, or contour surface, of a field defined by one or more spherical primitives:

$$\begin{aligned} \mathcal{S} &= \left\{ P \in \mathbf{R}^3 \mid F(P) = iso \right\} \\ F(P) &= \sum_{i=1}^{n} f_i(P) \\ f_i(P) &= \begin{cases} -k_i r + k_i e_i + 1 & \text{if} 0 \leq r \leq e_i \\ \frac{1}{4}\left[k_i(r - e_i) - 2\right]^2 & \text{if} e_i < r \leq R_i \\ 0 & \text{elsewhere} \end{cases} , \end{aligned} \tag{5}$$

where $r = d(P, S_i)$, $R_i = e_i + \frac{2}{k_i}$ and d is the Euclidean distance to the primitive's center [Tsingos *et al.*, 1995]. The f_i spherical field functions are thus parameterized by:

- The radius e_i of the sphere S_i created by a single primitive such that $f_i(e_i) = iso$;
- The stickiness k_i that defines the blending properties of the surface.

To speed-up the subsequent computations and to prevent the surface fitting method described below from diverging, we fix the stickiness k_i to 5, which means one less parameter to optimize for each implicit surface, and take iso to be 1. This offers sufficient precision, without letting the primitives become too small, in which case we would need a large number of them to obtain the final fitting solution. Thus, there are four parameters left to optimize for each primitive: e_i, x_i, y_i ,z_i, where (x_i, y_i, z_i) are the 3-D coordinates of the primitive's center.

4.3 Fitting the Implicit Surface

To compute $\mathcal{S}$ so that it approximates the data as well as possible, we look for a set of primitives that minimizes the energy E:

$$\begin{aligned} E &= \frac{1}{N}\left(\sum_{j=1}^{N}(f(P_j) - iso)^2\right) \\ &+ \frac{1}{M}\left(\alpha_1 \sum_{i=1}^{M} e^{+\beta_1 e_i}\right) \\ &+ \frac{1}{M}\left(\alpha_2 \sum_{i=1}^{M}\left[(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2\right]\right) , \end{aligned} \tag{6}$$

where M is the number of implicit surfaces and N the number of 3–D points. The first term of E forces the surface to approximate the points. The other two terms prevent the surface primitives from wildly increasing in size and from moving too far away from the center of mass. Without these terms, we would obtain large tangent surfaces. In this work, we use $\alpha_1 = 0$ and $\alpha_2 = 0.01$.

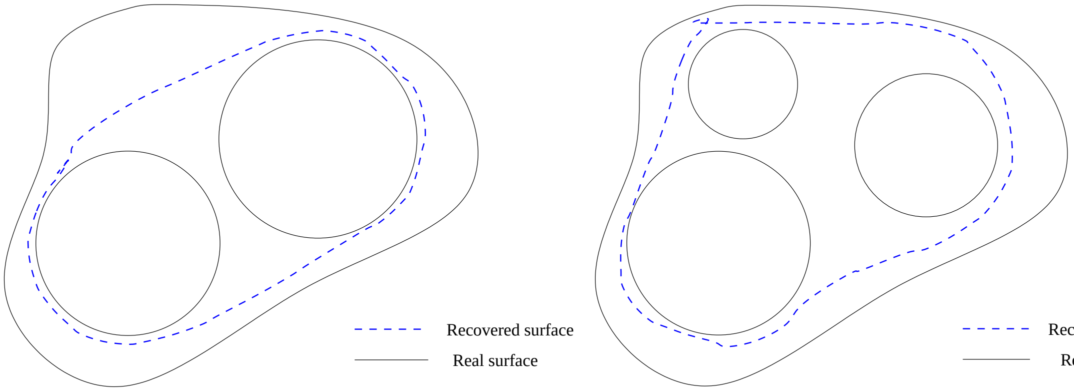


Figure 9: Splitting procedure. The implicit surface with the worst fit to the surface points, that is the largest C_i as defined by Eq. 7, is divided into two new ones that are then reoptimized.

To find the optimal set of primitives we are looking for, we implemented the following automated subdivision scheme, which is inspired by earlier methods [Bittar *et al.*, 1995, Tsingos *et al.*, 1995] and based on the local properties of the field function. We first place one spherical primitive at the center of mass, defined by:

$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i \,, \quad y_c = \frac{1}{N}\sum_{i=1}^{N} y_i \,, \quad z_c = \frac{1}{N}\sum_{i=1}^{N} z_i \quad ,$$

where N is the total number of data points, and x_i ,y_i, z_i are the coordinates of the ith 3-D point. The e parameter is the variance of the cloud defined as:

$$e = \frac{1}{N}\sum_{i=1}^{N} \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2}$$

We then recursively split the primitives based on a criterion C_i computed by summing the contributions of the m_i points that are inside the sphere of influence of the primitive S_i:

$$C_i = \frac{1}{m_i}\left(\sum_{j=1}^{m_i} (f(P_j) - iso)^2\right) \quad . \tag{7}$$

As shown in Fig. 9, the surface primitive with the largest C_i is subdivided first. This makes sense because this primitive is the one with the largest first term in Eq. 6 and, therefore, is the one whose area of influence corresponds to the zone of the surface where the reconstruction is worst. Each split involves subdividing the primitive into six new ones that are positioned on one axis in a symmetrical manner, so as to yield a shape as close as possible to the original. We then reoptimize the parameters of all the primitives to minimize the full criterion of Eq. 6.

Due to the variation in density of the data and the possible surface gaps this entails, the algorithm described above has a tendency to “cover” these gaps by introducing spurious primitives, thus allowing invalid rotations to be treated as valid. To alleviate this problem, after the final subdivision, we clean up the final result as follows. We remove all spherical primitives whose radius is larger than 1.0, as this would obviously include the entire cloud of data of unit quaternions, whose center is further away than 1.0 from the center of mass of the cloud, and whose density of point included in the primitive is small with respect to total density. Finally, we minimize again the criterion of Eq. 6, without subdivision of the primitives this time. This removes the primitives that grow through the holes but may still leave some that simply cover the holes, as will be seen in Section 5. This could be addressed by introducing negative implicit surfaces, which can be used to model real holes in the data. The difficulty, however, is to decide whether they correspond to truly forbidden areas or, simply, to missing data. In other words, in the presence of a suspected hole in the data, one should conduct additional motion capture sessions to confirm that the apparently forbidden area truly is unreachable.

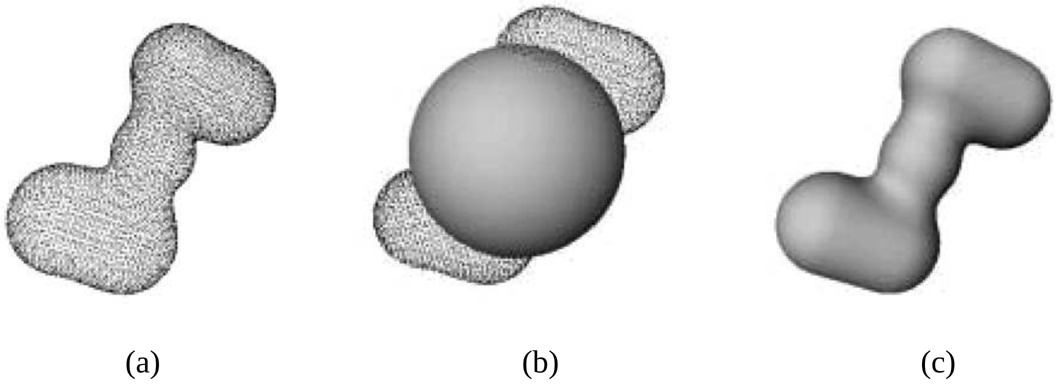
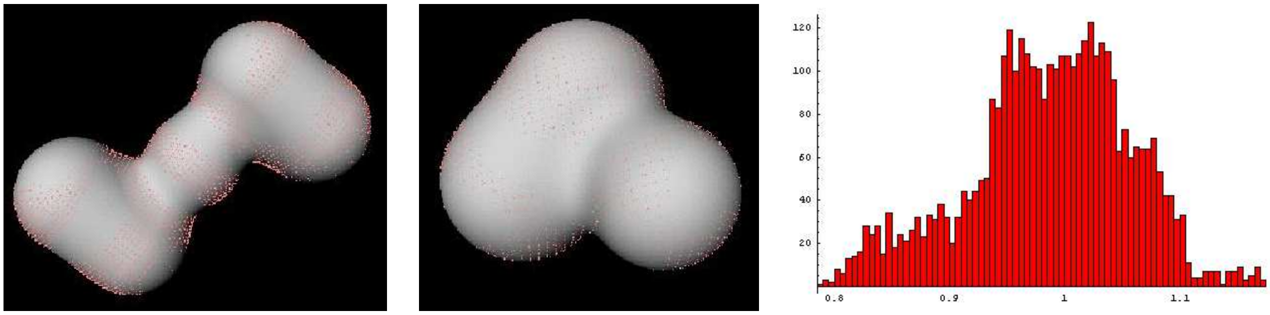


Figure 10: Reconstruction of a synthetic object. (a) Synthetic 3D point cloud. (b) Initialization of the algorithm with an implicit surface in the center of mass. (c) Final result of the reconstruction with 5 implicit surfaces

We have tested the validity of the subdivision algorithm by converting synthetic objects to a cloud of 3-D points, and, then, reconstructing them. The results are depicted by Figs. 10 and 11. In Fig. 11(c), we histogram the value of the energy E of Eq. 6 for the N points used to derive the shape shown in Fig. 11(b). Note that the histogram is correctly centered around the value of $iso = 1$.

4.4 Approximating the Shoulder Data

In short, we have developed a method goes through the following steps:

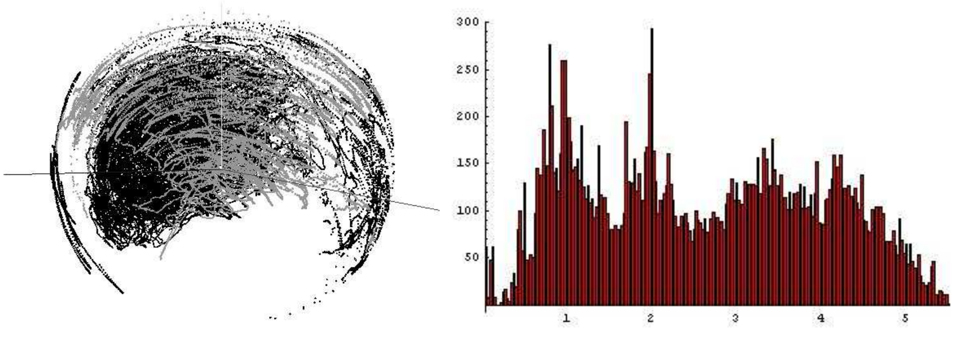


(a) (b) (c)

Figure 11: Two complete reconstructions. (a) Reconstructed shape of Fig. 10 with superposed 3-D points. (b) Reconstruction of another synthetic object. (c) Histogram of the value of the energy E of Eq. 6 evaluated at the points belonging to object (b).

1. Extract the surface points from the volumetric cloud of data;
2. Position an initial spherical primitive at the center of mass of the surface points;
3. Recursively subdivide this primitive and measure the error based on the distance from the surface points to the current spherical primitives.
4. Repeat step 3 until the error stabilizes.

We have applied this approach to approximating the collected shoulder data depicted by Fig. 6(a,b). The extracted surface points are shown in Fig. 6(c), and the resulting fitted implicit surface in Fig. 6(d).



(a) (b)

Figure 12: Comparing two subjects. (a) In black, the data of the subject we have been using for this study. In grey, the data corresponding to a second subject. (b) Histogram of values of the field function of Eq. 6 for the grey points.

To illustrate the relative insensitivity of these measurements across subjects, we have gathered shoulder motion data for a second subject. In Fig. 12(a), we overlay the two sets of 3–D points. Visual inspection in 3–D shows that they superpose well. This is confirmed by histogramming the value of the energy E of Eq. 6, that was derived using the data acquired with the first subject, for the data corresponding to the second one. As shown in Fig. 6(b), for the the majority of the points the value of E is greater than 1, which means that they are inside the "reachable" volume. Predictably, there are more points outside the surface, that is points for which E is smaller than 1, than in the synthetic case depicted by Fig. 11(c). However, outside of the implicit surface, the value of E decreases as the square of the distance to it. Thus, most of the points for which the value of E is smaller than 1 are in fact still very close to the surface.

5 From Implicit Surfaces to Joint Limits

In this section, we show how our results can be used both for animation and tracking purposes. Since each possible orientation is represented as a 3–D point with a clear and unambiguous metric inherited from the three-sphere, the mutual distance between individual orientations can be defined in a mathematically rigorous way. Consequently, given the implicit surface of the previous section that bounds the volume of acceptable rotations, we can not only perform consistency checks on the acceptability of a proposed or measured joint orientation but also orthogonally project invalid orientations onto the surface. The quaternion field method thus provides a new and unique method for establishing requirements and constraints for automated or semi-automated joint animations that can be used to reject or readjust proposed orientations to lie within a particular, experimentally verified, joint parameter space.

5.1 Animation constraints

In our laboratory's existing animation libraries, the ball-and-socket joints are implemented in terms of quaternions [Baerlocher and Boulic, 2000]. The orientation of such joints is decomposed into swing and twist components. The swing component, or angular motion, is limited by a spherical polygon. In other words, the sphere itself is centered at the joint—the shoulder, in our specific case—and the boundaries of this spherical polygon limit the possible orientations of the upper arm. The allowable twist component can then be expressed as a function of the swing one.

To take advantage of this existing implementation, we converted our implicit surface representation to the spherical polygon one. This implies sampling the surface of a sphere, validating each point with respect to the implicit surface, and constructing the spherical polygon from all valid points. This approach works but is somewhat cumbersome because converting to the spherical polygon representation entails a loss of precision that is almost inevitable when going from a continuous representation to a discrete one. It also forces us to decompose each motion into swing and twist.

We therefore implemented a much more promising approach to enforcing joint limits that directly uses the implicit surface of Section 4.3 to enforce the limits. In this manner, we need neither to discretize, nor to decompose each motion into swing and twist components.

In the examples shown in Figs. 13 and 14, an animator manually specified the motions depicted by the top two rows without worrying about joint limits. The resulting motions, shown both from the front and the side, are not humanly possible. We then used our implicit surfaces to project each invalid position to the closest valid one, resulting in the realistic motions depicted by the two bottom rows of Figs. 13 and Figs. 14. The corresponding animation sequences are available as mpeg movies on the web at the addresses shown in the Figures.

The projection of an invalid posture to a valid one is illustrated by Fig. 15. It is implemented by first determining which sphere S_i of the implicit surface has the most influence on the point *p* representing the invalid orientation. We then calculate the distance *d* from the surface of S_i to *p*, that is the distance from center of S_i to *p*, minus the radius of S_i. We trace a ray emanating from *p*, along the gradient of the total field function of Section 4.2. We determine the point *p'* on that ray that is at distance *d* from *p;* this point should be inside the implicit surface. From here on, we proceed by dichotomy, taking the middle point of the segment [*p*, *p'*], testing whether it is inside the implicit surface, and if not, we continue performing dichotomy—in the direction of the implicit surface if the point is outside, and in the reverse direction if the point is inside—until we find the intersection point.

Fig. 16 depicts a case where the implicit surface we use fails to appropriately constrain the motion. As shown in Fig. 6(b), there is a gap in our motion capture data that corresponds to positions of the arm behind the back that are clearly impossible. As discussed in Section 4.3, the problem comes from the fact that, in our existing implementation, the implicit surface fitting has a tendency to “cover” the gaps, thus allowing invalid rotations to be treated as valid. In future work, we will investigate the use of negative implicit surface to carve out regions that are void of data and solve this problem.

5.2 Video-based tracking

The technique described in Section 5.1 for animation purposes is equally applicable to video-based tracking. In earlier work, we have developed an approach to modeling and tracking human bodies using video from two or more cameras as input [Plänkers and Fua, 2001]. We extract stereo-data from the images and, at each time step, adjust the position of the body model to fit the data as closely as possible by minimizing an objective function. Here too, we use an implicit surface formalism to represent the model’s skin because it allows us to define this objective function so that it is both differentiable and computable without search.

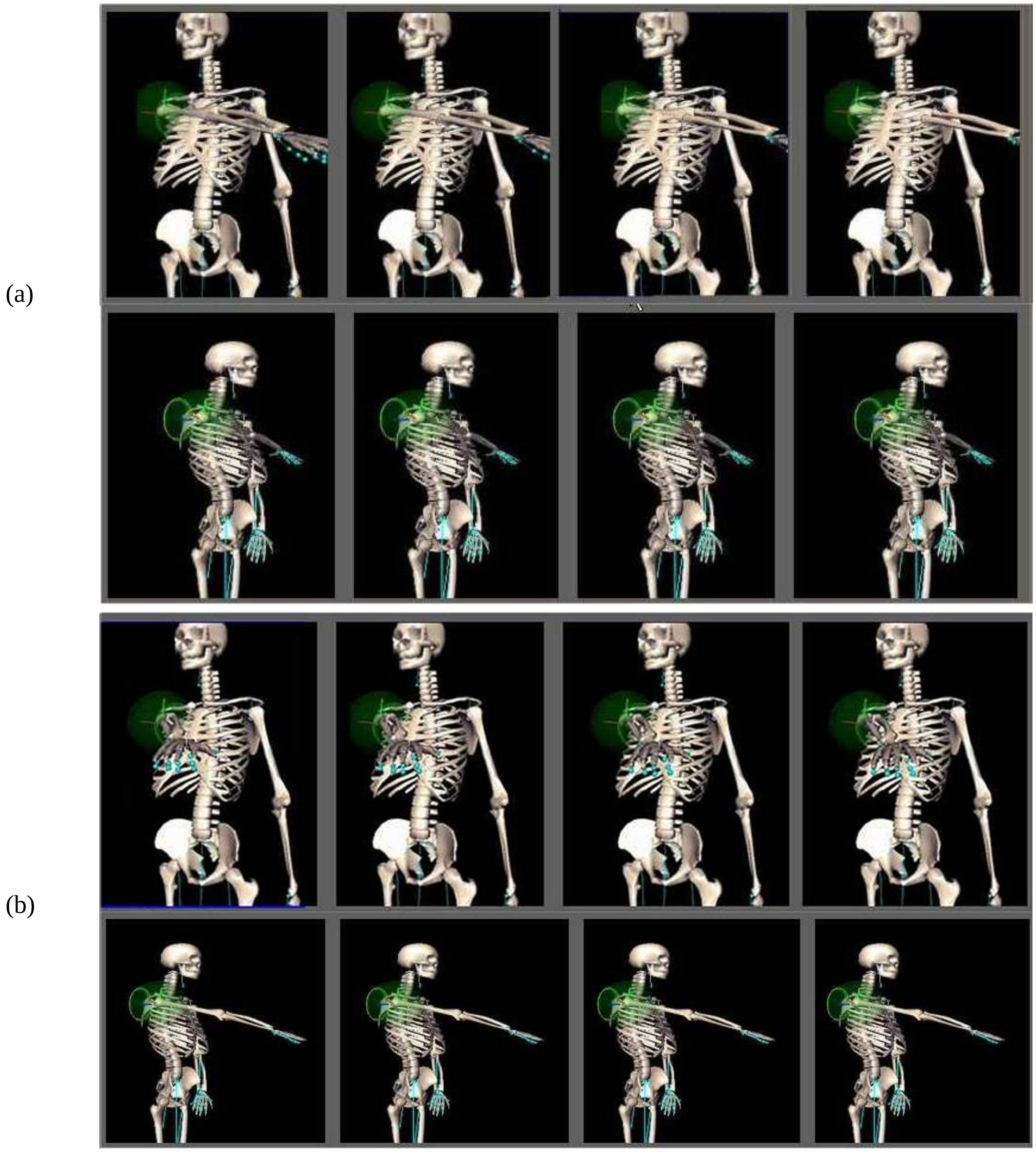


Figure 13: Animation without and with constraints. (a) Manually specified motion without constraints seen from the front (top) and from the side (bottom). Note that the arm penetrates the thorax, which would hurt a lot. (b) The same motion after joint limits have been enforced. The green sphere represents the joint limits and replaces the spherical polygon of Fig. 1(b).

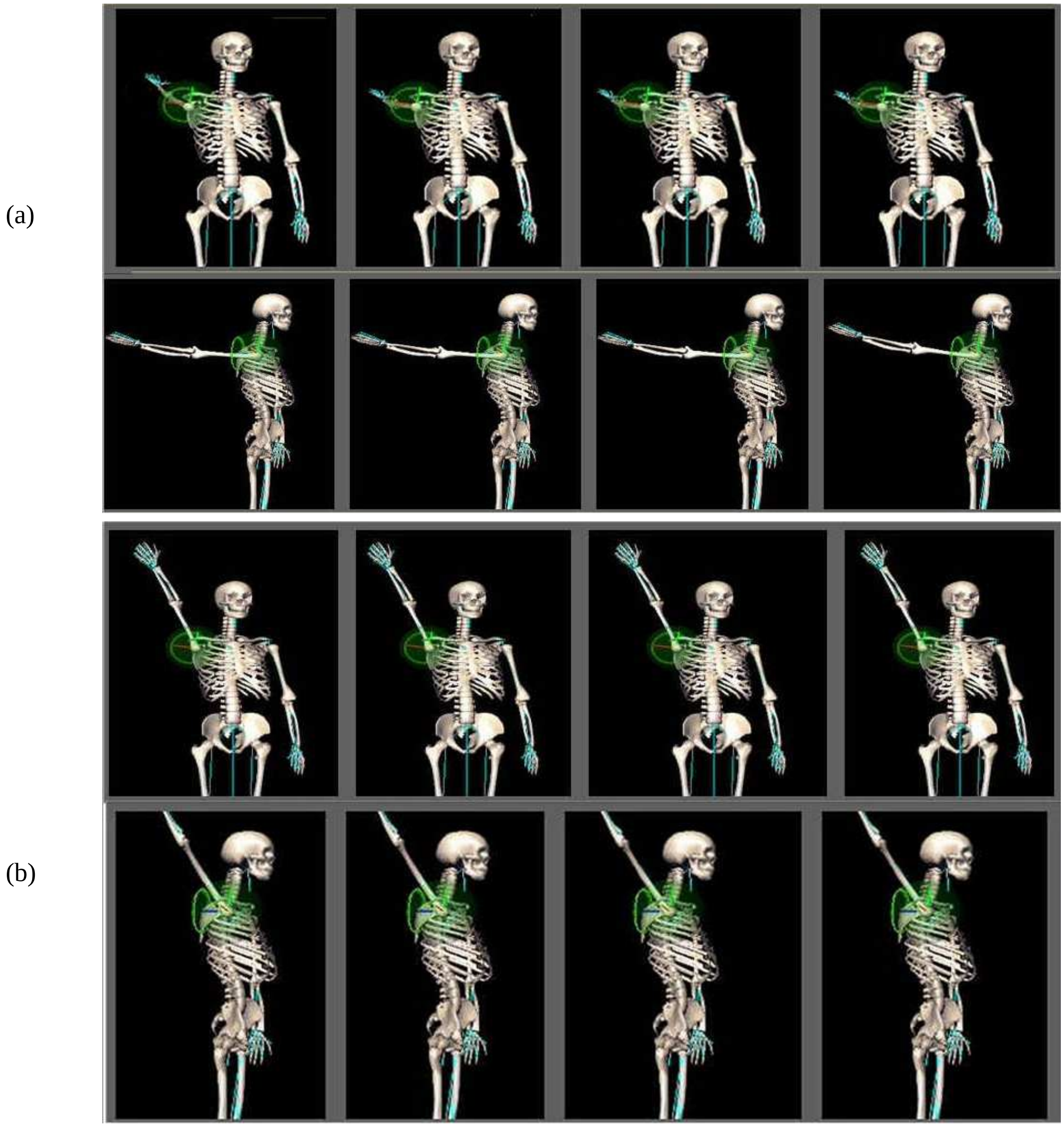


Figure 14: Another example of an unconstrained and constrained motion. (a) The arm reaches too far backwards when no joint-limits constrain the motion. (b) The joint limits have been enforced, the motion is limited to its natural boundary posture.

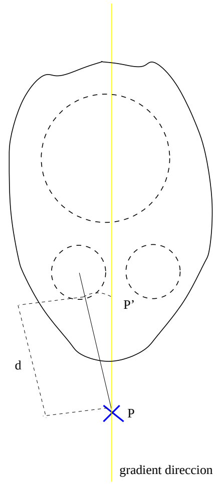


Figure 15: Closest valid orientation determination along the gradient of the field function of Equation 6.

Because the model we use has many degrees of freedom, it can in some cases match the data accurately while assuming a position that no real human could. However, by imposing the joint limit constraints, the model switches to a much more plausible pose. An example is shown in Figure 17: Tracking the arm without enforcing joint-limit constraints results in an impossible twist at the shoulder level. In this case, the constraints were imposed after the fact but the formalism proposed in this paper will make it natural to incorporate them directly into the tracking process: Constraint satisfaction or violation is expressed in terms of the differentiable function of Eq. 6 and can therefore be naturally incorporated into our optimization framework using standard robotics-style constraint satisfaction techniques [Baerlocher and Boulic, 2000].

6 Conclusion

We have developed a promising technique for automatically determining ball-and-socket joint limits that does not require extensive data collection and nevertheless produces a useful output both within a short time and in an immediately re-usable format. In this way, we are in position to greatly improve our ability to correctly animate and track human bodies.

In this work, our aim was to develop a method for retrieving joint limits for ball-and-socket joints, such as our simplified shoulder or the hip joint, rather than to propose a fully realistic shoulder model. In future work, however, we intend to consider a more sophisticated model that explicitly incorporates the shoulder, scapula, and clavicle joint. By measuring the motion of each independently, for example by constraining the motion of the scapula, we expect our approach to extend naturally to this more complex case. This should result in an even more effective set of constraints for robust tracking and realistic animation.

We will also improve our surface fitting procedure, so as to increase the method's robustness with respect to gaps in the data. These are simply the areas where there are no data because

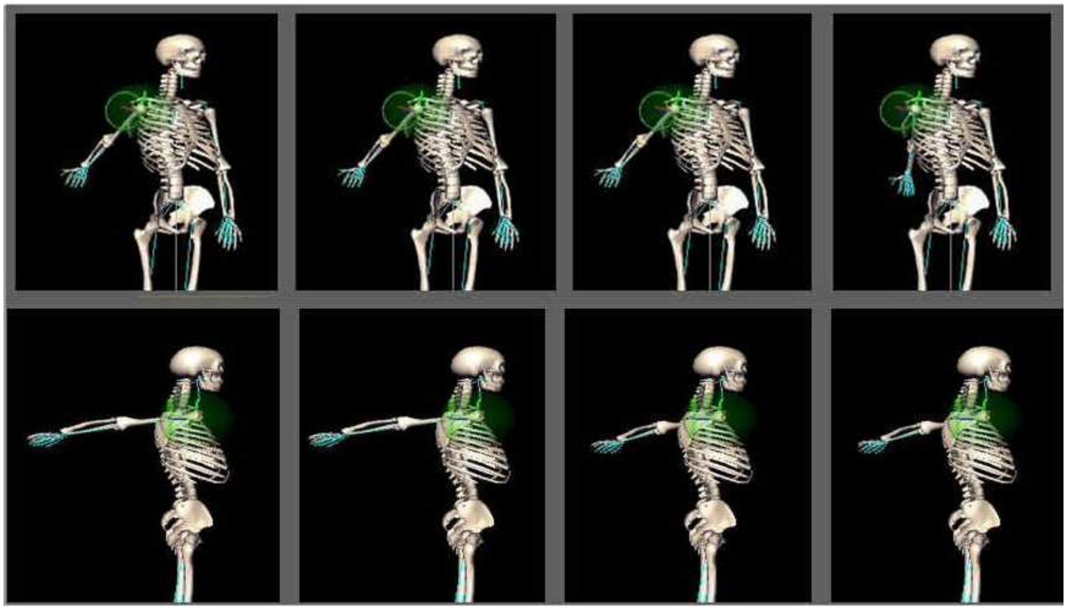


Figure 16: Insufficient flexion constraint in the back region: Because our implicit surface covers data gaps, we allow orientations that should have been invalidated.

they are invalid positions and show up as holes within the extracted data points. They create problems such as those shown in the previous section and will be handled by introducing negative implicit surface to carve out regions that are void of data.

We will ultimately develop a tool that provides joint limits in an intuitive manner and that takes into account all rotation components, without any need for motion decomposing. Because it is generic, this method is generalizable to all the joints in the body and should therefore prove valuable for both animation and tracking applications.

Acknowledgements

The authors would like to thank Paolo Baerlocher, Ralf Plaenkers, Richard Lengagne, Eric Bittar and especially Amaury Aubel for their valuable contribution to this work.

References

[Baerlocher and Boulic, 2000] P. Baerlocher and R. Boulic. Parameterization and range of motion of the ball-and-socket joint. In *Proceedings of Avatars 2000*, 2000.

[Biedenharn and Louck, 1981] Lawrence C. Biedenharn and James D. Louck. *Angular Momentum in Quantum Physics: Theory and Application*. Addison-Wesley, 1981. Volume in the Encyclopedia of Mathematics and Its Applications.

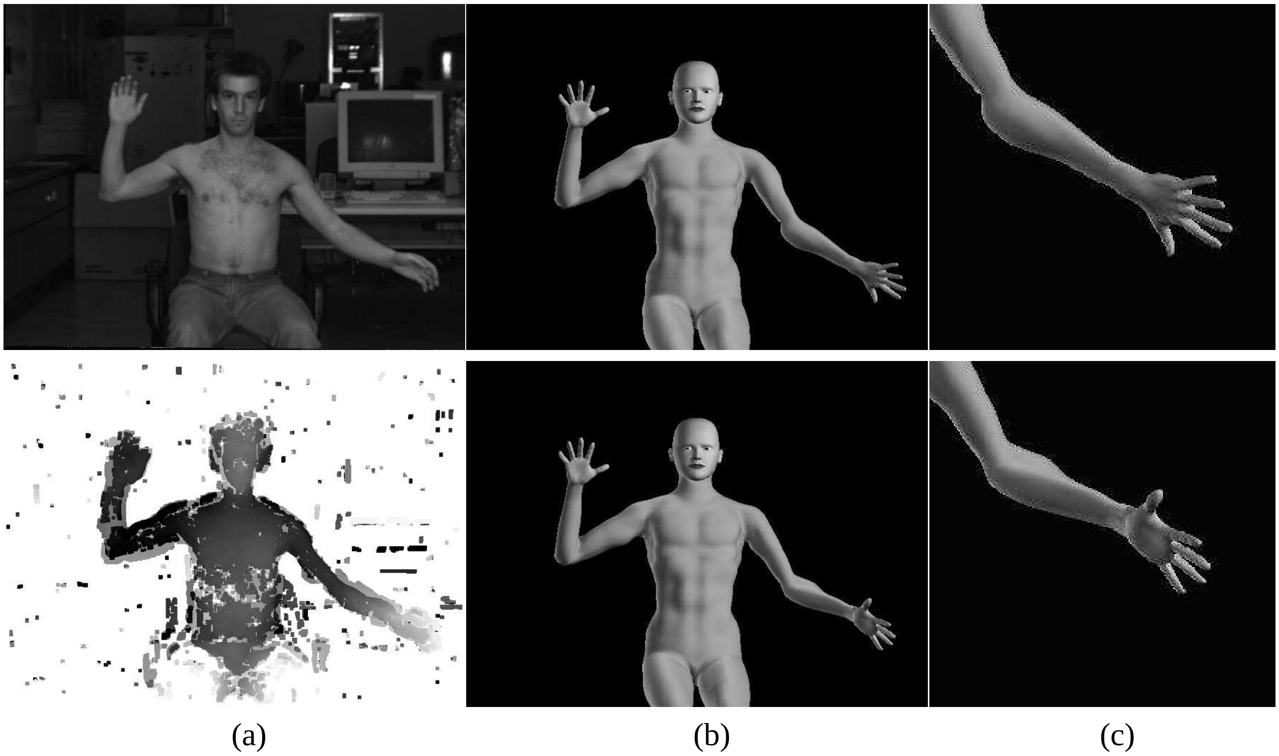


Figure 17: Using constraints in a tracking context. Top row: (left) One image of a sequence of stereo pairs in which the subject lifts his arm. (center) Tracking the left arm without shoulder constraints results in an impossible shoulder twist. (right) Close-up view. Bottow row: (left) Disparity map corresponding to the image above. (center) Imposing the angular constraints yields the correct pose, as opposed to the wrong pose shown above. (right) Close-up view.

[Bittar *et al.*, 1995] E. Bittar, N. Tsingos, and M.P. Gascuel. Automatic reconstruction of unstructured 3D data: Combining medial axis and implicit surfaces. *Computer Graphics Forum*, 14(3):457–468, 1995.

[Bloomenthal, 1990] Jules Bloomenthal. Calculation of reference frames along a space curve. In Andrew Glassner, editor, *Graphics Gems*, pages 567–571. Academic Press, Cambridge, MA, 1990.

[Bobick, 1998] N. Bobick. Rotating objects using quaternions. *Game Developer*, 2, Issue 26, 1998.

[Bregler and Malik, 1998] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR*, 1998.

[Engin and Tumer, 1989] A.E. Engin and S.T. Tumer. Three-dimensional kinematic modeling of the human shoulder complex. *Journal of Biomechanical Engineering*, 111:113–121, 1989.

[Engin and Tumer, 1993] A.E. Engin and S.T. Tumer. Improved dynamic model of the human knee joint and its response to imact loading on the lower leg. *Journal of Biomechanical Engineering*, 115(2):137–143, 1993.

[Gavrila, 1999] D.M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1), January 1999.

[Grassia, 1998] F.S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.

[Hanson and Ma, 1995] A. J. Hanson and H. Ma. Quaternion frame approach to streamline visualization. *IEEE Trans. on Visualiz. and Comp. Graphics*, 1(2):164–174, June 1995.

[Hanson, 1998a] A. J. Hanson. Constrained optimal framings of curves and surfaces using quaternion gauss maps. In *Visualization*, pages 375–382. IEEE Computer Society Press, 1998.

[Hanson, 1998b] A.J. Hanson. Quaternion Gauss Maps and Optimal Framings of Curves and Surfaces. Technical Report 518, Indiana University Computer Science Department, 1998.

[Johnston and Smidt, 1969] R. Johnston and G. Smidt. Measurement of hip joint motion during walking. *Journal of Bone and Joint Surgery*, 51(A):1083–1094, 1969.

[Korein, 1985] J.U. Korein. *A Geometric Investigation Of Reach*. MIT Press, Cambridge, 1985.

[Maurel, 1998] W. Maurel. 3d modeling of the human upper limb including the biomechanics of joints, muscles and soft tissues, 1998.

[Meskers *et al.*, 1998] C.G.M. Meskers, , H.M. Vermeulen, J.H. de Groot, F.C.T. Van der Helm, and P.M. Rozing. 3d shoulder position measurements using a six-degree-of-freedom electromagnetic tracking device. *Clinical Biomechanics*, 13:280–292, 1998.

[Moeslund and Granum, 2001] T.B. Moeslund and E. Granum. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81, March 2001.

[Muraki, 1991] S. Muraki. Volumetric shape description of range data using “Blobby Model”. *Computer Graphics*, 25(4):227–235, 1991.

[Opalach and Maddock, 1995] A. Opalach and S. C. Maddock. An overview of implicit surfaces. In *Introduction to Modelling and Animation Using Implicit Surfaces*, pages 1.1–1.13, 1995. Leeds, UK. Course Notes No 3.

[Plänkers and Fua, 2001] R. Plänkers and P. Fua. Articulated soft objects for video-based body modeling. In *ICCV*, Vancouver, Canada, July 2001.

[Rehg *et al.*, 2003] J. M. Rehg, D. D. Morris, and T. Kanade. Ambiguities in Visual Tracking of Articulated Objects using 2–D and 3–D Models. *International Journal of Robotic Research*, 2003. This issue.

[Shoemake, 1985] K. Shoemake. Animating rotation with quaternion curves. In *Computer Graphics*, volume 19, pages 245–254, 1985. Proceedings of SIGGRAPH 1985.

[Tolani *et al.*, 2000] D. Tolani, N. Badler, and J. Tallier. A kinematic model of the human arm using triangular bézier spline surfaces, 2000.

[Tsingos *et al.*, 1995] N. Tsingos, E. Bittar, and M.P. Gascuel. Implicit surfaces for semi-automatic medical organs reconstruction. In *Computer Graphics International'95*, pages 3–15, Leeds, UK, 1995.

[Wang *et al.*, 1998] X. Wang, M. Maurin, F. Mazet, N. De Castro Maia, K. Voinot, J.P. Verriest, and M. Fayet. Three-dimensional modelling of the motion range of axial rotation of the upper arm. *Journal of Biomechanics*, 31(10):899–908, 1998.

[Watt and Watt, 1992] A. Watt and M. Watt. Advanced animation and rendering techniques, 1992.