Automatic Joint Parameter Estimation from Magnetic Motion Capture Data

James F. O'Brien Bobby Bodenheimer Gabriel J. Brostow Jessica K. Hodgins

College of Computing and Graphics, Visualization, and Usability Center
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, GA 30332-0280
e-mail: [obrienj|bobbyb|brostow|jkh]@cc.gatech.edu

Abstract

This paper describes a technique for using magnetic motion capture data to determine the joint parameters of an articulated hierarchy. This technique makes it possible to determine the limb lengths, joint locations, and sensor placement for a human subject without external measurements. Instead, the joint parameters are inferred with high accuracy from the motion data acquired during the capture session. The parameters are computed by performing a linear least squares fit of a revolute joint model to the input data. A hierarchical structure can also be determined in situations where the topology of the articulated model is not known. We present the results of running the algorithm on human motion capture data, as well as validation results obtained with data from a simulation and a wooden linkage of known dimensions.

Keywords: Animation, Motion Capture, Kinematics, Parameter Estimation, Joint Locations, Articulated Figure, Articulated Hierarchy.

1 Introduction

Motion capture has proven to be an extremely useful technique for animating human and human-like characters. Because the basic motion is specified in real-time by the subject being captured, motion capture provides a unique solution for applications where animations with the characteristic qualities of human motion must be generated quickly. Real-time capture techniques can be used to create immersive virtual environments for entertainment and training applications. Motion capture data retains many of the subtle elements of a performer's style making possible digital performances where the motion capture subject's unique style is recognizable in the final product.

Although motion capture has many advantages and commercial systems are improving rapidly, the technology also has some drawbacks. Both optical and magnetic systems suffer from sensor noise and require careful calibration [7]. Additionally, measurements of the subject such as limb lengths or the offsets between the sensors and the joints are often required. This information is usually gathered by measuring the subject in a reference pose but this hand measurement is tedious and prone to error. It is also impractical for such applications as location-based entertainment, where the delay and physical con-

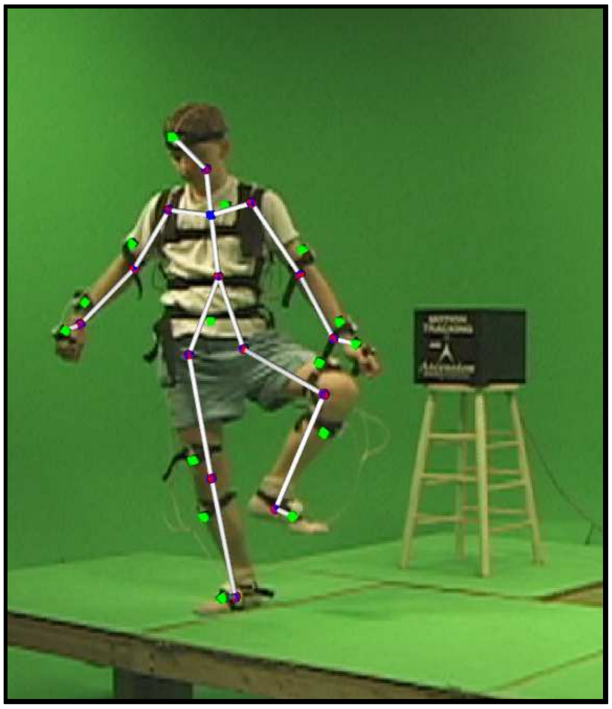


Figure 1: **Test subject and generated model.** The image shows a subject wearing the motion capture equipment during a capture session with the skeletal model generated automatically from the acquired motion capture data superimposed.

Submitted to Graphics Interface 2000.

tact with a technician would be unacceptable.

The algorithm described in this paper addresses this problem by computing the joint locations for an articulated hierarchy automatically from the global transformation matrices of the individual bodies. We take motion data acquired with a magnetic system and determine both the locations of the subject's joints and the sensor locations relative to the joints without any external measurement. The technique does not impose any constraints on the sensor positions beyond those necessary for accurate capture, nor does it require the subject to pose in any particular configuration. The only requirement on the data is that it must exercise all degrees of freedom of the joints for the technique to return an unambiguous answer. Figure 1 shows a human subject wearing magnetic motion capture sensors and the skeletal model that was generated from the motion data in an automatic fashion.

Intuitively, the algorithm proceeds by examining the sequences of transformation data generated by pairs of sensors and determines a pair of points, one in the coordinate system of each sensor, that remain collocated throughout the sequence. If the two sensors were attached to a pair of objects that were joined by a revolute joint, then a single point, the center of the joint, will fulfill this criteria. Errors such as sensor noise and the fact that human joints are not perfect revolute joints, will prevent an exact solution. The algorithm solves for a best fit solution and computes the residual error that describes how well two bodies "fit" together. This metric makes it possible to infer the body hierarchy directly from the motion data by building a minimum spanning tree treating the residuals as edge weights between the body parts.

In the following sections, we describe related work in the fields of biomechanics and robotics and our method for computing the joint locations from motion data. We present the results of processing human motion capture data, as well as validation results using data from a simulation and from a wooden linkage of known dimensions.

2 Background

The problem of determining a system's kinematic parameters from the motion of the system has been widely studied in the fields of biomechanics [18, 19] and robotics [12]. Biomechanicists are interested in this problem because the joints play a critical role in understanding both the mechanics of the human body and the dynamics of human motion. However, human joints are not ideal revolute joints and therefore do not have a fixed center of rotation. Even joints like the hip which are relatively close approximations to mechanical ball and socket joints have laxity and variations due to joint loading that cause changes in the center of rotation during movement. Instead, the parameter that is often measured in biomechanics is the instantaneous center of rotation, which is defined as the point of zero velocity during infinitesimally small motions of a rigid body.

To compute the instantaneous center of rotation, biomechanists put markers on each limb and use measurements from different configurations of the limbs. To reduce the error in this measurement, multiple markers are used on each joint and a least squares fit is used to filter the redundant marker data [4]. Spiegelman and Woo [21] proposed a method for planar motions, and this was extended to general motion by Veldpaus *et al.* [24]. The latter algorithm uses multiple markers on a body measured at two instants in time to establish the center of rotation. Because of practical limitations on the number of markers that can be used, algorithms have been developed to calculate the optimal placement of these markers [6, 11].

We are primarily concerned with creating animation rather than scientific studies of human motion, and as a result our goals differ from those of researchers in the biomechanics community. In particular, because the recorded motion will be used to drive an articulated skeleton, we need joint centers that are a reasonable approximation over the entire sequence of motion as opposed to an instantaneous joint center that is more accurate but describes only a single instant of motion.

The biomechanical literature also provides insight into the errors inherent in a joint estimation system and provides an upper bound on the accuracy that we can expect. Because the joints of the human body are not revolute, the articulated models used in animation are an inherent approximation to true human motion. Using five male subjects with pins inserted in their tibia and femur, Lafortune *et al.* [14] found that during a normal walk cycle the joint center of the knee compressed and pulled apart by an average of 7 mm, moved front-to-back by 14.3 mm, and side-to-side by 5. 6 mm. Another source of error arises because we cannot attach the markers directly to the bone. Instead, they are attached to the skin or clothing of the subject. Ronsky and Nigg reported up to 3 cm of skin movement over the tibia during ground contact in running [17].

Roboticists are also interested in similar questions because they need to calibrate physical devices. An articulated robot may be built to precise specifications, but there will be differences between the nominal parameters and those of the actual unit. Furthermore, because a robot is made of physical materials that are subject to various types of deformation, there may be additional degrees of freedom in the actual unit that were not part of the design specification. Both of these types of differences can have a significant effect on the accuracy of the unit and compensating for them often requires that they be measured [12]. Taking these measurements directly can be extremely difficult so researchers have developed various automatic calibration techniques.

The calibration techniques that are relevant to the research presented here infer these parameters indirectly by measuring the motion of the actual robot. Some of these techniques require that the robot perform specific actions such as exercising each joint in isolation [26, 16] or that it assume a particular set of configurations [13, 3], and are therefore not easily adapted for use with human performers. Other methods allow calibration from an arbitrary set of configurations but focus explicitly on the relationship between the control parameters and the end-effector. Although our technique fits into the general framework described by Karan and Vukobratovic for estimating linear kinematic parameters from arbitrary motion [12], the techniques are not identical because of differences in the goals and constraints of animation versus robotics. In particular, we are interested in information about the entire body rather than only the person's end-effectors. We also can take advantage of the position and orientation information provided by the magnetic motion sensors whereas robotic calibration methods are generally limited to the information provided by joint sensors (that may themselves be part of the set of parameters being calibrated) and position markers on the end-effector.

3 Methods

For a system of m rigid bodies, let $\mathcal{T}^{i\to j}$ be the transformation from the i-th body's coordinate system to the coordinate system of the j-th body ($i, j \in [0..m-1]$). The index $\omega \notin [0..m-1]$, is used to indicate the world coordinate system so that $\mathcal{T}^{i\to\omega}$ is the global transformation from the i-th body's coordinate system to the world coordinate system.

A transformation, $\mathcal{T}^{i\to j}$, consists of an additive, length 3 vector component, $\mathbf{t}^{i\to j}$, and a multiplicative, 3×3 matrix component, $\mathbf{R}^{i\to j}$. We will refer to $\mathbf{t}^{i\to j}$ as the translational component of $\mathcal{T}^{i\to j}$, and to $\mathbf{R}^{i\to j}$ as the rotational component of $\mathcal{T}^{i\to j}$, although in general $\mathbf{R}^{i\to j}$ may be any invertible 3×3 matrix transformation.

A point, $\mathbf{x}^i$, expressed in the i-th coordinate system may then be transformed to the j-th coordinate system by

$$\mathbf{x}^j = \mathbf{R}^{i\to j}\mathbf{x}^i + \mathbf{t}^{i\to j}. \tag{1}$$

A transformation from the i-th coordinate system to the j-th coordinate system may be inverted so that given $\mathcal{T}^{i\to j}$, $\mathcal{T}^{j\to i}$ may be computed by

$$\mathbf{R}^{j\to i} = (\mathbf{R}^{i\to j})^{-1} \tag{2}$$

$$\mathbf{t}^{j\to i} = (\mathbf{R}^{i\to j})^{-1}(-\mathbf{t}^{i\to j}), \tag{3}$$

where $(\cdot)^{-1}$ indicates matrix inverse.

The subsequent algorithms require the global transformations of the m rigid bodies, $\mathcal{T}^{i\to\omega}$, as input. If the available data is in the form of relative or hierarchical transformations, they may be converted to global form by compositing the transformations. Given $\mathcal{T}^{i\to j}$ and $\mathcal{T}^{j\to q}$ with $i, j, q \in [\omega, 0..m-1]$, then $\mathcal{T}^{i\to q}$ is computed by

$$\mathbf{R}^{i\to q} = \mathbf{R}^{j\to q}\mathbf{R}^{i\to j} \tag{4}$$

$$\mathbf{t}^{i\to q} = \mathbf{R}^{j\to q}\mathbf{t}^{i\to j} + \mathbf{t}^{j\to q}. \tag{5}$$

Because in general the bodies are in motion with respect to each other and the world coordinate system, the transformations between coordinate systems change over time. We will assume that the motion data is sampled at n discrete moments in time called frames, and use $\mathcal{T}_k^{i\to j}$ to refer to the value of $\mathcal{T}^{i\to j}$ at frame $k \in [0..n-1]$. The rotational and translational components of $\mathcal{T}_k^{i\to j}$ are $\mathbf{R}_k^{i\to j}$ and $\mathbf{t}_k^{i\to j}$ respectively.

An articulated hierarchy is described by the topological information indicating which bodies are connected to each other, and by geometric information indicating the locations of the connecting joints. The topological information takes the form of a tree[1] with a single body located at its root and all other bodies appearing as nodes within the tree as shown in Figure 2. When referring to directions relative to the arrangement of the tree, the *inboard* direction is towards the root, and the *outboard* direction is away from the root. Thus for a joint connecting two bodies, i and j, the parent body, j, is the inboard body and the child, i, is the outboard body. Similarly, a joint which connects a body to its parent is that body's inboard joint and a joint connecting the body to one of its children is an outboard joint. All bodies have at most one inboard joint but may have multiple outboard joints.

The hierarchy's topology is defined using a mapping function, $P(\cdot)$, that maps each body to its parent body so that $P(i) = j$ will imply that the j-th body is the immediate parent of the i-th body in the hierarchical tree. The object, $\tau \in [0..m-1]$, with $P(\tau) = \omega$ is the root object. To simplify discussion, we will temporarily assume that $P(\cdot)$ is known *a priori*. Later, in Section 3.3, we will show how $P(\cdot)$ may be inferred when only the $\mathcal{T}^{i\to\omega}$'s are known.

The geometry of the articulated hierarchy is determined by specifying the location of each joint in the coordinate frames of both its inboard body and its outboard

[1] We discuss the topological cycles created by loop joints in Section 5.

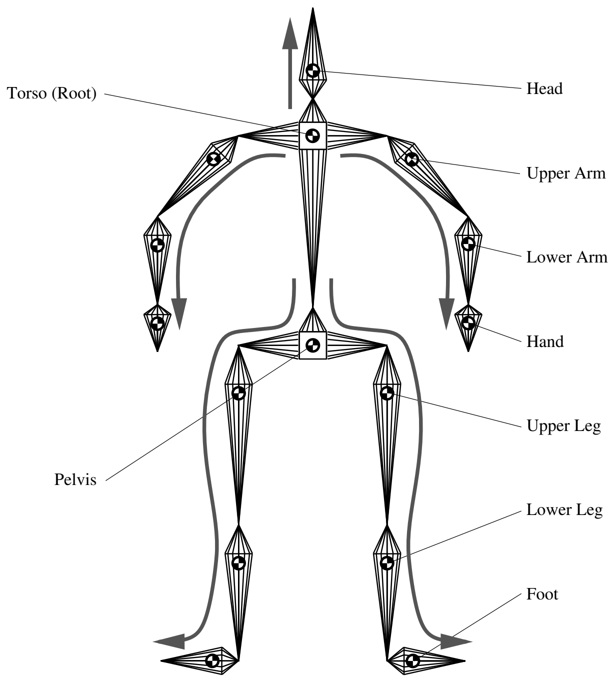


Figure 2: **Example of an articulated hierarchy** that could be used to model a human figure. The torso is the root body and the grey arrows indicate the outboard direction. For rendering, the skeleton model shown here would be replaced with more realistic graphical model.

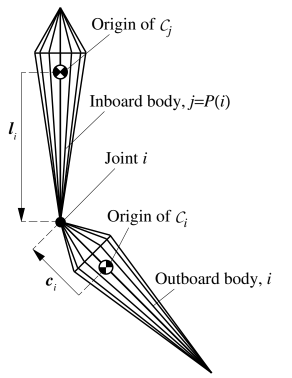


Figure 3: **Joint diagram** showing the location of the rotary joint between bodies i and $j = P(i)$. The location of the joint is defined by a vector displacement, $\mathbf{c_i}$, relative to the coordinate system of body i, and a second vector displacement, $\mathbf{l_i}$, in the coordinate system of body j.

body. Because each body has a single inboard joint, we will index the joints so that the i-th joint is the inboard joint of the i-th body as shown in Figure 3.

Let $\mathbf{c}_i$ refer to the location of the i-th joint in the i-th body's (the joint's outboard body) coordinate system, and let $\mathbf{l}_i$ refer to the location of the i-th joint in the $P(i)$-th body's (the inboard body's) coordinate system (see Figure 3). The transformation of equation (1) that goes from the i-th coordinate system to its parent's, $P(i)$, coordinate system can then be re-expressed in terms of the joint locations, $\mathbf{c}_i$ and $\mathbf{l}_i$, and the rotation at the joint, $\mathbf{R}^{i \to P(i)}$, so that

$$\begin{aligned} \mathbf{x}^{P(i)} &= \mathbf{R}_k^{i \to P(i)}(\mathbf{x}^i - \mathbf{c}_i) + \mathbf{l}_i && (6) \\ &= \mathbf{R}_k^{i \to P(i)}\mathbf{x}^i - \mathbf{R}_k^{i \to P(i)}\mathbf{c}_i + \mathbf{l}_i. && (7) \end{aligned}$$

3.1 Finding Joint Locations

The general transformation given by equation (1) applies to any arbitrary hierarchy of bodies. When the bodies are connected by rotary joints, the relative motion of two connected bodies must satisfy a constraint that prevents the joint between them from coming apart. Comparing equation (6) with equation (1) shows that while rotational terms are the same, the translational term of equation (1) has been replaced with the constrained term, $-\mathbf{R}_k^{i \to P(i)}\mathbf{c}_i + \mathbf{l}_i$. Using equation (6) to transform the location of $\mathbf{c}_i$ to the $P(i)$-th coordinate system will identically yield $\mathbf{l}_i$, and equation (6) naturally enforces the constraint that the joint stay together.

Although the input transformations for each of the body parts do not contain any explicit information about joint constraints, if the motion was created by an articulated system then it should be possible to express the same transformations hierarchically using equation (6) and an appropriate choice of $\mathbf{c}_i$ and $\mathbf{l}_i$ for each of the joints. Thus for each pair of parent and child bodies, $i \neq \tau$ and $j = P(i)$, there should be a $\mathbf{c}_i$ and $\mathbf{l}_i$ such that equation (1) and equation (6) become equivalent and

$$\begin{aligned} &\mathbf{R}_k^{i \to P(i)}\mathbf{x}^i + \mathbf{t}_k^{i \to P(i)} = \\ &\qquad \mathbf{R}_k^{i \to P(i)}\mathbf{x}^i - \mathbf{R}_k^{i \to P(i)}\mathbf{c}_i + \mathbf{l}_i \qquad (8) \end{aligned}$$

for all $k \in [0..n-1]$. After canceling the common terms, equation (8) becomes

$$\mathbf{t}_k^{i \to P(i)} = -\mathbf{R}_k^{i \to P(i)}\mathbf{c}_i + \mathbf{l}_i \qquad (9)$$

for all $k \in [0..n-1]$. Later, it will be more convenient to work with the global transformations. By applying $\mathcal{T}^{P(i) \to \omega}$ to both sides of equation (9) and simplifying the result, we get

$$\mathbf{R}_k^{i \to \omega}\mathbf{c}_i + \mathbf{t}_k^{i \to \omega} = \mathbf{R}_k^{P(i) \to \omega}\mathbf{l}_i + \mathbf{t}_k^{P(i) \to \omega} \qquad (10)$$

for all $k \in [0..n-1]$. Equation (10) has a consistent geometric interpretation: the location of the joint in the outboard coordinate system, $\mathbf{c}_i$, and the location of the joint in the inboard coordinate system, $\mathbf{l}_i$, should transform to the same location in the world coordinate system; in other words, the joint should stay together.

Equation (10) can be rewritten in matrix form as

$$\mathbf{Q}_k^{i\to P(i)}\mathbf{u}_i = \mathbf{d}_k^{i\to P(i)}. \tag{11}$$

where $\mathbf{d}_k^{i\to P(i)}$ is the length 3 vector given by

$$\mathbf{d}_k^{i\to P(i)} = -(\mathbf{t}_k^{i\to\omega} - \mathbf{t}_k^{P(i)\to\omega}), \tag{12}$$

$\mathbf{u}_i$ is the length 6 vector

$$\mathbf{u}_i = \begin{bmatrix} \mathbf{c}_i \\ \mathbf{l}_i \end{bmatrix}, \tag{13}$$

and $\mathbf{Q}_k^{i\to j}$ is the 3×6 matrix

$$\mathbf{Q}_k^{i\to j} = \left[(\mathbf{R}_k^{i\to\omega}) \; (-\mathbf{R}_k^{P(i)\to\omega}) \right]. \tag{14}$$

Assembling equation (11) into a single linear system of equations gives for all $0..n-1$ frames

$$\begin{bmatrix} \mathbf{Q}_0^{i\to P(i)} \\ \vdots \\ \mathbf{Q}_k^{i\to P(i)} \\ \vdots \\ \mathbf{Q}_{n-1}^{i\to P(i)} \end{bmatrix} \begin{bmatrix} \mathbf{c}_i \\ \mathbf{l}_i \end{bmatrix} = \begin{bmatrix} \mathbf{d}_0^{i\to P(i)} \\ \vdots \\ \mathbf{d}_k^{i\to P(i)} \\ \vdots \\ \mathbf{d}_{n-1}^{i\to P(i)} \end{bmatrix}. \tag{15}$$

The matrix of $\mathbf{Q}$'s we will denote by $\widehat{\mathbf{Q}}$ and is $3n \times 6$; the matrix of $\mathbf{d}$'s is $3n \times 1$. The linear system of equations (15) can be used to solve for the joint location parameters, $\mathbf{c}_i$ and $\mathbf{l}_i$.

Unless the input motion data consists of only two frames of motion, $\widehat{\mathbf{Q}}$ will have more rows than columns and the system will, in general, be over-constrained. Nonetheless, if the motion was generated by an articulated model, then an exact solution will exist. Realistically, limited sensor precision and other sources of error will prevent an exact solution, so a best fit solution must be found instead.

Despite the fact that the system will be over-constrained, it may be simultaneously under-constrained if the input motions do not span the space of rotations. In particular, if two bodies connected by a joint do not rotate with respect to each other, or if they do so but only about a single axis, then there will be no unique answer. In the case where they are motionless with respect to each other then any location in space would be a solution. Similarly, if their relative rotations are about a single axis, then any point on that axis could serve as the joint's location. For reasons of numerical accuracy, in either of these cases the desired solution is chosen to be the one closest to the origin of the inboard and outboard body coordinate frames.

The technique of solving for a least squares solution using singular value decomposition is well suited for this type of problem [20]. The matrix $\widehat{\mathbf{Q}}$, is decomposed into the product of three matrices

$$\widehat{\mathbf{Q}} = \mathbf{U}\mathbf{W}\mathbf{V}^\top, \tag{16}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{W}$ is the diagonal matrix of singular values of $\widehat{\mathbf{Q}}$. Equation (15) can then be solved by

$$\begin{bmatrix} \mathbf{c}_i \\ \mathbf{l}_i \end{bmatrix} = \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^\top \begin{bmatrix} \mathbf{d}_0^{i\to P(i)} \\ \vdots \\ \mathbf{d}_k^{i\to P(i)} \\ \vdots \\ \mathbf{d}_{n-1}^{i\to P(i)} \end{bmatrix}. \tag{17}$$

Because $\mathbf{W}$ is diagonal, computing its inverse is trivially accomplished by setting $\mathbf{W}^{-1}$ to a diagonal matrix whose elements are the reciprocals of those in $\mathbf{W}$. If the system is under- or nearly under-constrained, then some of the diagonal elements of $\mathbf{W}$ will be zero or near zero, and the corresponding columns of $\mathbf{V}$ are directions in which the input data has not constrained the joint location. When this occurs, the corresponding elements of $\mathbf{W}^{-1}$ should be set to zero, and the indices noted for later use in Section 3.2.

The algorithm for computing the singular value decomposition is beyond the scope of this paper. It is commonly discussed in many scientific computing texts, and most numerical analysis packages contain an implementation of the algorithm (see for example [2] and [20]). Other least squares methods, such as solving the "normal equations," may be easier to implement and slightly faster but they are less robust, which is particularly important if the system is under- or nearly under-constrained. When under-constrained, many less robust least squares techniques will be unable to handle the resulting singular matrix and fail with an exception. Worse, if the system is nearly under-constrained then otherwise small numerical errors in the data may drive the joint location for a single-axis joint to an undesirably distant location along the axis.

Once a solution has been determined, how well the solution fits the input data may be measured by computing the residual vector to equation (15):

$$\begin{bmatrix} \mathbf{e}_0^{i\to P(i)} \\ \vdots \\ \mathbf{e}_k^{i\to P(i)} \\ \vdots \\ \mathbf{e}_{n-1}^{i\to P(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_0^{i\to P(i)} \\ \vdots \\ \mathbf{Q}_k^{i\to P(i)} \\ \vdots \\ \mathbf{Q}_{n-1}^{i\to P(i)} \end{bmatrix} \begin{bmatrix} \mathbf{c}_i \\ \mathbf{l}_i \end{bmatrix} - \begin{bmatrix} \mathbf{d}_0^{i\to P(i)} \\ \vdots \\ \mathbf{d}_k^{i\to P(i)} \\ \vdots \\ \mathbf{d}_{n-1}^{i\to P(i)} \end{bmatrix}. \tag{18}$$

The vector $\mathbf{e}_k^i$ is the translation difference between the input data and the value given by equation (6) at frame k for a given $\mathbf{c}_i$ and $\mathbf{l}_i$. The average error distance,

$$\varepsilon_i = \frac{\sqrt{\sum_{k=0}^{n-1} \| \mathbf{e}_k^{i\to P(i)} \|^2}}{n}, \tag{19}$$

gives a measure of the overall quality for the fit of the joint.

3.2 Single-Axis Joints

If a joint rotates about two or more non-parallel axes, there will be enough information to resolve the location of the joint center as described above. As mentioned previously, however, if the joint rotates about a single axis then a unique joint center does not exist, and our algorithm will only return a point on the line representing the axis of rotation.

We are able to detect that a joint is a single-axis joint and to determine that axis from the input data by examining the singular values of $\widehat{\mathbf{Q}}$ from equation (15). If one of the singular values of $\widehat{\mathbf{Q}}$ is near zero, i.e., if $\widehat{\mathbf{Q}}$ is rank deficient, then that joint is a single-axis joint, or at least in the input motion it rotates only about a single axis. The first three components of the corresponding column vector of $\mathbf{V}$ from the singular value decomposition are the joint axis in the inboard coordinate frame, and the second three are the axis in the outboard coordinate frame

If the original motion only rotated about the one axis then, it seems reasonable that subsequent motions should be similarly constrained. If this is not the desired case, then the joint location determined by equation (15) can be used for general rotations and if the results are incorrect then the user may adjust the location along the axis.

3.3 Determining the Body Hierarchy

In the previous sections, we have assumed that the hierarchical relationship between the bodies given by the parent function, $P(\cdot)$, is known. In some instances, however, it may be desirable to determine a suitable hierarchy automatically by inferring it from the input transformation matrices. Our algorithm does this by finding the parent function that minimizes the sum of the ε_i's for all the joints in the hierarchy.

The problem of finding the optimal hierarchy is equivalent to finding a minimal spanning tree. Each body can be thought of as a node, joints are the edges between them, and the joint fit error, ε_i, is the weight of the edge. The hierarchy can then be determined by evaluating the joint error between all pairs of bodies, selecting a root node, τ, and then constructing the minimal spanning tree. See [5] for example algorithms. Because the main computational cost for this process is likely to be the evaluation of the joint fit errors, the joint fit error for the edge implied by $P(i) = j$ and by $P(j) = i$ are the same.

3.4 Removing the Residual

Once we have determined the locations of the joints, we can use this information to construct a model that approximates the dimensions of the subject. This model can then be used to playback the original motion data. Unless the residual errors on the joint fits were all near zero, the motion will cause the joints of the model to move apart from each other during playback in a fashion that is typical of unconstrained motion capture data. If, however, we use the inferred joint locations to create an articulated model with kinematic joint constraints and then play back the motion through this model the joints will stay together. Playing back motion capture data by applying only the rotations to an articulated model is common practice, the difference here is that the model itself has been generated from the motion data. Essentially, we have projected the motion data onto a parametric model and then used the fit to discard the residual.

4 Results

The technique described above was tested on a rigid body dynamic simulation of a human containing 48 degrees of freedom. The simulation was an active simulation similar to the ones described by Hodgins *et al.* [10], and was moved so that all the degrees of freedom were exercised. The algorithm correctly computed the limb lengths within the limits of numerical precision (errors less than 10^{-6} m) and determined the correct hierarchy.

Our method was next tested in a magnetic motion capture environment. Magnetic motion capture systems are frequently noisy, and our Ascension system [1] has a resolution of about 4 mm. To establish a baseline for the amount of noise present in the environment, two sensors were rigidly attached 56.5 cm apart and moved through the capture space. The results of this experiment are shown in Figure 4. There is a scale factor in converting from units the motion capture system reports to centimeters, and we calculated this scale factor based on the mean

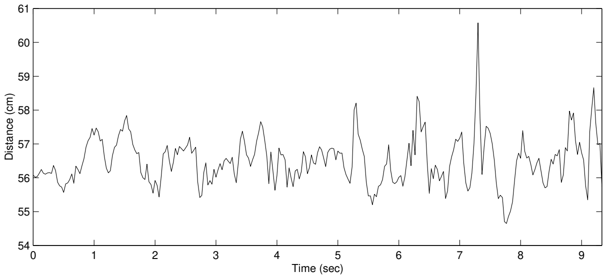
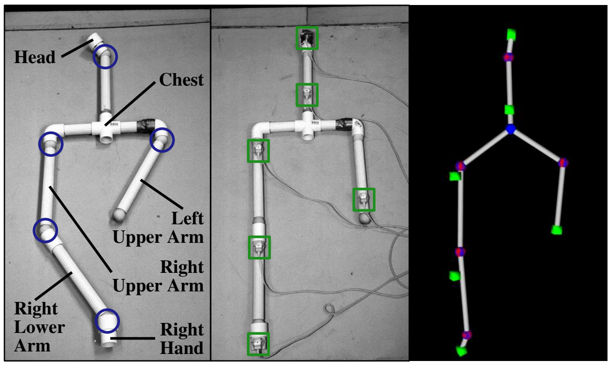


Figure 4: **Calibration data** showing the distance between two markers attached rigidly to one another and moved through the capture space. The data is scaled to have a mean of 56.5 cm, and the standard deviation is 0.7 cm.

of this data set. If the sensors are rigidly attached and not moved, then the data is much less noisy.

To test the algorithm on something less complicated than biological joints, we constructed a wooden mechanical linkage with five three-degree-of-freedom revolute joints. The linkage is shown in Figure 5. Six sets of data were captured in which all the degrees of freedom were exercised. Before Set 6 was captured, the marker positions were moved to evaluate the robustness of the method to changes in marker locations. The results are shown in Table 1 along with the measured values of the joint-to-joint distances. The maximum error across all trials is 1.1 cm and the hierarchy was computed correctly for each of the trials. Another way of evaluating the fit is to examine the residual vectors from the least squares process. The norms of the residual vectors for the best fit (Set 1, Right Shoulder) and the worst fit (Set 6, Left Shoulder) are shown in Figures 6 and 7, respectively. The right-hand graph has an asymmetric distribution because it is the distribution of an absolute value. We regard these results as very good because the error is on the order of the resolution of the sensors.

The best test case, of course, is to verify that we can estimate the limb lengths of people. This task is more difficult because human joints are not simple mechanical linkages. To provide a basis for comparison, we measured the limb lengths of our test subjects. As mentioned previously, this process is inexact and prone to error, but it does provide a plausible estimate with which to compare. We measured limb lengths from bony landmark to bony landmark to provide repeatability and consistency to our measurements. For example, the upper leg of the subjects was measured as the distance from the top of the greater trochanter of the femur to the lateral condyle of the tibia. Because the head of the femur extends upward and inward into the innominate, it is clear that this mea-



(A) (B) (C)

Figure 5: **Wooden mechanical linkage.** (A) Labels indicate the terms that we used to refer to the body parts, and purple circles highlight the joint locations. (B) The motion capture sensors (highlighted green squares) have been attached to the linkage. (C) The model computed automatically from the motion data using our algorithm. The joints are shown with blue spheres, and the sensors with green cubes. Links between joints are indicated with grey cylinders.

surement is inaccurate by a few centimeters. Nonetheless, as the greater trochanter is the only palpable area at the upper end of the femur, this measurement is the best available. This difficulty in obtaining accurate hand measurements is on of the primary reasons that we chose to develop our automatic technique.

Our test subjects performed two different sets of motions for capture. The first set we refer to as the “exercise” set, and it consists of the subjects attempting to move every joint essentially in isolation, to generate a full range of motion for each joint. Thus the routine consists of a set of discrete motions such as rolling the head around on the neck, bending at the waist, high-stepping, lifting one leg and waving it about, lifting the arms and waving them about, bending the elbows and the wrists, etc. This exercise set mimics the way we gathered data for the mechanical linkage. The second set of motions captured are referred to as the “walk” sets, and consists of the subjects trying to move as many degrees of freedom at once as they can in a walk motion. This routine is perhaps best described as a “chicken” walk, consisting of highly exaggerated leg movements coupled with bending the waist and waving the arms about.

A male test subject performed the two different types of motion and they were processed with the algorithm. The results of the limb length calculations are shown in Tables 2 and 3. As expected, the residual errors for a human are much larger than for the mechanical linkage.

	Meas.	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Neck — Left Shoulder	39.0	39.4	38.8	39.8	39.1	39.1	40.1
Neck — Right Shoulder	39.7	39.8	39.8	40.3	40.0	39.9	40.3
Between Shoulders	34.3	34.3	33.7	34.5	34.3	34.3	34.8
Right Upper Arm	28.6	29.2	29.0	28.8	28.9	29.0	29.1
Left Upper Arm	31.4	31.5	31.7	31.9	31.5	31.1	31.2
		Δ 1	Δ 2	Δ 3	Δ 4	Δ 5	Δ 6
Neck — Left Shoulder		-0.4	0.2	-0.8	-0.1	-0.1	-1.1
Neck — Right Shoulder		-0.1	-0.1	-0.6	-0.3	-0.2	-0.6
Between Shoulders		0.0	0.6	-0.2	0.0	0.0	-0.5
Right Upper Arm		-0.6	-0.4	-0.2	-0.3	-0.4	-0.5
Left Upper Arm		-0.1	-0.3	-0.5	-0.1	0.3	0.2

Table 1: **A comparison** of measurements and calculated limb lengths for six data sets of the mechanical linkage. The units are cm and the columns labeled Δ show the difference in measured and calculated values. Joint names follow the analogy with human physiology used in Figure 5(A).

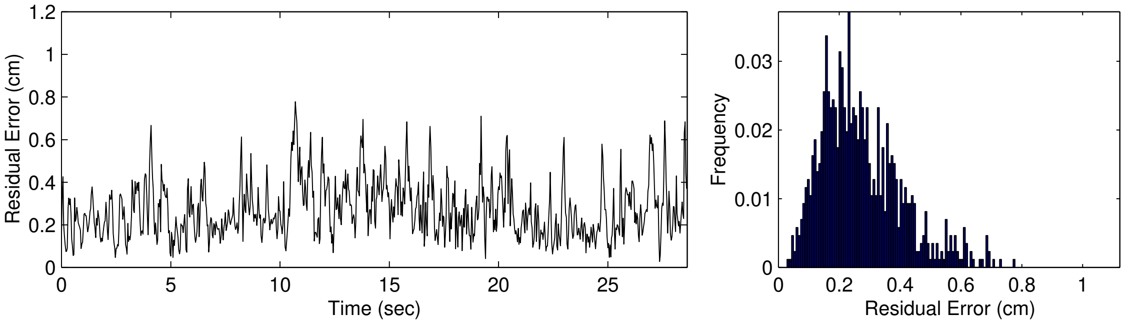


Figure 6: **Residual errors of the right shoulder joint** for the data from Set 1 for the mechanical linkage (table 1). In the left graph, the magnitude of the residual vector is shown. In the right graph, the distribution of the frequency of the magnitudes is shown.

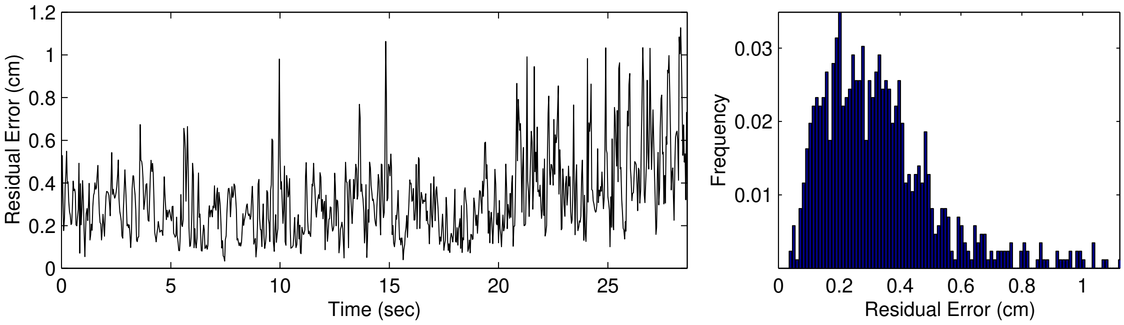


Figure 7: **Residual errors of the left shoulder joint** for the data from Set 6 for the mechanical linkage (table 1). In the left graph, the magnitude of the residual vector is shown. In the right graph, the distribution of the frequency of the magnitudes is shown.

A representative example is shown in Figure 8. For this subject, the maximum difference between measured and calculated values is 4.1 cm, and occurs at the left upper arm during one of the exercise sets. The mean of the differences between calculated and measured values is less than one centimeter for every limb except the upper arms, where it is 1.4 and 2.2 cm for the right and left arms, respectively. The algorithm consistently finds a longer length for the left upper arm than what we measured, and it is possible that the difference is due to an error in the value measured by hand. However, the shoulder joint is poorly approximated by a revolute joint: an accurate biomechanical rigid-body model would have at least seven degrees of freedom [23, 22]. Thus, it is not surprising that the worst fit occurs on the shoulder.

The same motions were repeated with a female test subject and the results are shown in Table 4. The largest difference between calculated and measured values is 2.4 cm and again occurs for the the left upper arm. The algorithm also finds a longer length for the left upper arm than we measured. The maximum error is less than that for the male test subject, but there is less consistency among the results for the female test subject. The mean of the differences between the calculated and measured values is greater than one centimeter for the right lower leg, left upper leg, and left upper arm.

The system also computed a hierarchy for each of the trials. For all of the "exercise" trials for both the male and female subject the computed hierarchy was correct. However, the results from the "walk" data were less satisfactory. Out of the five "walk" trials, for three of them the algorithm improperly made one of the upper legs a child of the other, instead of the pelvis. We believe that this occurred because the pelvis sensor is mounted on the system's battery pack that the subject wears on his/her hip. The sway of the hip causes motion in the sensor that could have contributed to this error. The limb length results we have reported are, of course, for the correct hierarchy assignments.

In addition to the joint measurements we have reported, our algorithm determines information for joints, such as between the chest and pelvis, that model the bending of the torso but which are gross approximations to the way the spine bends. Our algorithm reports limb lengths for these joints within the torso, and they are generally consistent with the dimensions of the torsos of the subjects. However, because we have no reasonable way measuring these lengths for comparison, we have omitted them from our presentation of the results. The locations computed for these joints can be seen in figure 1 and the animations that accompany this paper.

Finally, it is worth noting that the current process is quite fast. On an SGI O2 with a 195 MHz R10000 processor, it takes less than 4 seconds to process 45 seconds of motion data for 16 sensors with the hierarchy specified, and less than 14 seconds to compute the hierarchy.

5 Discussion and Conclusions

This paper presents an automatic method for computing the limb lengths, joint locations, and sensor placement from magnetic motion capture data. The method produces results accurate to the resolution of the sensors for data that was recorded from a mechanical device constructed with revolute joints. For data recorded from a human subject, the accuracy of the results is consistent with the estimates in the biomechanics literature for the error introduced by approximating human joints as revolute and assuming that the skin does not move with respect to the bone.

In a production animation environment, measuring and calibrating a performer is one of the most tedious and expensive parts of the process. Because this algorithm runs very quickly, it provides a rapid way to accomplish the calibration for magnetic motion capture systems. Detecting marker slippage or correcting for marker slippage are additional complications in the motion capture pipeline. By looking for large changes in the joint residual, this technique provides a rapid way of determining if a marker slipped during a particular recorded segment and allowing the segment to be performed again while the subject is still suited with sensors.

The parameters computed by this method can be used to create a digital version of a particular performer by matching a graphical model to the proportions of the motion capture subject. The process does not require the subject to assume a particular pose or perform specific actions other than to fully exercise their joints. Therefore, the method could be incorporated into applications where explicit calibration is infeasible. A cleverly disguised "exercise" routine could be part of the pre-show portion of a location-based entertainment experience, for example.

The algorithm would also be of use in applications where the problem is fitting data to a graphical model with different dimensions than the motion capture subject. The algorithm presented here could be used in a pre-processing step to provide the best-fit limb lengths for the data and modify the data to have constant limb lengths. Then constraint-based techniques could be applied to adapt the resulting motion to the new dimensions of the graphical character.

We used magnetic motion capture data in our work because we had that type of system available, but the technique should work with optical data as well. Many optical

	Meas.	Exer. 1	Exer. 2	Exer. 3	Exer. 4	Δ 1	Δ 2	Δ 3	Δ 4
Right Lower Leg	40.0	40.8	40.9	42.2	42.5	-0.8	-0.9	-2.2	-2.5
Left Lower Leg	40.3	37.3	38.4	41.2	41.5	3.0	1.9	-0.9	-1.2
Right Upper Leg	41.6	41.5	42.1	42.9	42.2	0.1	-0.5	-1.3	-0.6
Left Upper Leg	43.2	41.4	41.8	43.2	43.0	1.8	1.4	0.0	0.2
Right Lower Arm	27.0	26.3	26.7	27.7	27.0	0.7	0.3	-0.7	0.0
Left Lower Arm	26.7	26.5	27.0	26.7	27.1	0.1	-0.3	-0.1	-0.4
Right Upper Arm	29.5	32.1	31.3	29.3	28.8	-2.6	-1.8	0.2	0.7
Left Upper Arm	29.5	33.7	32.9	30.1	29.9	-4.1	-3.4	-0.6	-0.4

Table 2: **A comparison** of measurements and calculated limb lengths for four data sets of a male test subject attempting to exercise each degree of freedom essentially in isolation. The units are cm and the columns labeled Δ show the difference in measured and calculated values for the appropriate set.

	Meas.	Walk 1	Walk 2	Walk 3	Δ 1	Δ 2	Δ 3
Right Lower Leg	40.0	40.7	40.3	38.9	-0.6	-0.3	1.1
Left Lower Leg	40.3	40.8	38.9	39.8	-0.4	1.4	0.5
Right Upper Leg	41.6	40.7	40.6	42.6	0.9	1.0	-1.0
Left Upper Leg	43.2	45.1	42.7	43.1	-1.9	0.5	0.1
Right Lower Arm	27.0	27.3	27.5	25.8	-0.3	-0.5	1.2
Left Lower Arm	26.7	26.2	24.9	25.6	0.5	1.7	1.0
Right Upper Arm	29.5	31	31.1	32.7	-1.4	-1.6	-3.2
Left Upper Arm	29.5	32.3	32.3	30.8	-2.7	-2.7	-1.3

Table 3: **A comparison** of measurements and calculated limb lengths for three data sets of a male test subject attempting to exercise all degrees of freedom simultaneously. The units are cm and the columns labeled Δ show the difference in measured and calculated values for the appropriate set.

	Meas.	Exer. 1	Walk 1	Walk 2	Δ_e 1	Δ_w 1	Δ_w 2
Right Lower Leg	36.8	39.1	38	38.1	-2.3	-1.2	-1.3
Left Lower Leg	36.5	37.6	37	37.4	-1.1	-0.5	-0.9
Right Upper Leg	42.2	42.9	43.3	42.2	-0.7	-1.1	0.0
Left Upper Leg	41.9	42.4	44.1	42.9	-0.5	-2.2	-1.0
Right Lower Arm	24.8	25.5	25.3	22.4	-0.7	-0.5	2.3
Left Lower Arm	24.8	25.1	24.8	23	-0.3	0.0	1.8
Right Upper Arm	27.6	27.5	27.5	28.7	0.2	0.1	-1.0
Left Upper Arm	27.6	28.5	30	29	-0.9	-2.4	-1.3

Table 4: **A comparison** of measurements and calculated limb lengths for four data sets of a female test subject. The column labeled "Exercise" denotes a performance attempting to exercise each degree of freedom in isolation, while the columns labeled "Walk" denote a performance attempting to exercise all degrees of freedom simultaneously. The units are cm and the columns labeled Δ show the difference in measured and calculated values for the appropriate set.

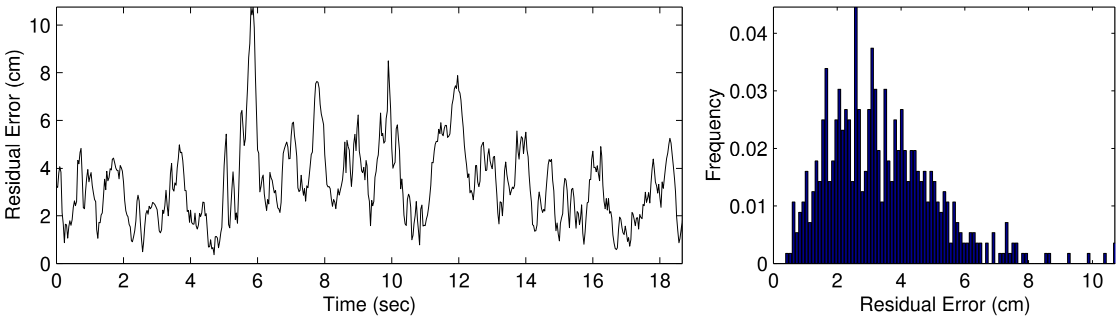


Figure 8: **Residual errors of the left shoulder** for the data from Walk 2 of the male test subject (table 3). In the left graph, the magnitude of the residual vector is shown. In the right graph, the distribution of the frequency of the magnitudes is shown. The scale of the residual vectors is larger than that of the residual vectors for Figures 6 and 7.

systems use techniques similar to those described by [17] to fit a transformation matrix to several position markers attached to a single limb. This transformation matrix provides translation and orientation data for each body part; our algorithm can then be applied to the data just as it was applied to data from the magnetic system. The noise properties of optical data are different from magnetic data, however, which may affect the quality of the fit.

Passive optical systems often have problems with marker identification because occlusion causes markers to appear to swap. For example, when the hand passes in front of the hip during walking, the marker on the hand and the one on the hip may become confused. If this happens, the marker locations may change relatively smoothly but the joint center of the inboard and outboard bodies for each marker will change discontinuously. This error should be identifiable when the data is processed allowing the markers to be disambiguated.

For relatively clean data, this algorithm can be used to extract the hierarchy automatically. Specifying the hierarchy is not burdensome for magnetic motion capture data because the markers are uniquely identified by the system. However, automatic identification of the hierarchy might be useful in situations where the connections between objects are dynamic such as pairs dancing or a subject manipulating an instrumented object.

We have assumed that the hierarchy is a strict tree and does not contain any cycles or loop joints such as the closed chain that is created when the hands are clasped together. If the hierarchy is known *a priori* then the location of a loop joint is found just as it is for any other joint. If the hierarchy is not known, then the method of Section 3.3 will not find cycles and the hierarchy it returns will be missing the additional joints required to close the loops. This problem could be detected by informing the user that a joint fit with a low error was not used in building the tree.

The algorithm we have described is statistically equivalent to fitting a parameterized model to a distribution. The revolute joint model that is commonly used for skeletal animation is linear, but more complex models that explicitly model the errors introduced by the non-revolute nature of the joints, the slippage of skin, or the noise distribution seen in the magnetic setup would be non-linear. Non-linear models have been used in the robotics literature to model elastic deformation of robot limb segments, joints that do not have a fixed center of rotation, and dynamic variation due to system inertial properties [15, 8, 25, 9, 12]. Reconstructing the motion based on the joint locations, as described in Section 3.4, is a first step towards identifying the components of the motion that are due to actual motion and those that are due to errors. The addition of more sophisticated models would allow us to separate the components of the data that are attributable to the motion of the subject from the components that are due to other sources. This separation would possibly allow very accurate data to emerge even from systems where the sensors are only loosely attached to the subject.

Acknowledgements

The authors would like to thank Victor Zordan for helping with the motion capture equipment and the use of his software. Christina De Juan also helped with various phases of the motion capture process. We also thank Len Norton for his assistance during the early stages of this project.

This project was supported in part by NSF NYI Grant No. IRI-9457621, Mitsubishi Electric Research Laboratory, and a Packard Fellowship. The first author was supported by a Fellowship from the Intel Foundation, and the second author by an NSF CISE Postdoctoral award. The motion capture system was purchased with funds from NSF Instrumentation Grant No. 9818287.

References

[1] Ascension Technology Corp., http://www.ascension-tech.com.

[2] K. E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons Inc., New York, NY, 1978.

[3] J.-H. Borm and C.-H. Menq. Experimental study of observability of parameter errors in robot calibration. In *Proceeding of the 1989 IEEE International Conference on Robotics and Automation*, pages 587–592. IEEE Robotics and Automation Society, 1998.

[4] J. H. Challis. A procedure for determining rigid body transformation paramters. *Journal of Biomechanics*, 28(6):733–737, 1995.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithims, fourth edition*. McGraw–Hill Book Company, 1991.

[6] J. J. Crisco, X. Chen, M. M. Panjabi, and S. W. Wolfe. Optimal markder placement for calculating the instantaneous center of rotation. *Journal of Biomechanics*, 27(9):1183–1187, 1994.

[7] B. Delaney. On the trail of the shadow woman: The mystery of motion capture. *IEEE: Computer Graphics and Applications*, 18(5):14–19, 1998.

[8] A. A. Goldenberg, X. He, and S. P. Ananthanarayanan. Identification of inertial parameters of a manipulator with closed kinematic chains. *IEEE: Transactions on Systems, Man, and Cybernetics*, 22(4):799–805, 1992.

[9] R. Gourdeau, G. M. Cloutier, and J. Laflamme. Parameter identification of a semi-flexible kinematic model for serial manipulators. *Robotica*, 14:331–319, 1996.

[10] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *SIGGRAPH '95 Conference Proceedings*, Annual Conference Series, pages 71–78. ACM SIGGRAPH, Aug. 1995. Held in Los Angeles, California, 6–11 August 1995.

[11] S. Holzreiter. Calculation of the instantaneous centre of rotation for a rigid body. *Journal of Biomechanics*, 24(7):643–647, 1991.

[12] B. Karan and M. Vukobratović. Calibration and accuracy of manipulation robot models – an overview. *Mechanism and Machine Theory*, 29(3):479–500, 1992.

[13] D. H. Kim, K. H. Cook, and J. H. Oh. Identification and compensation of robot kinematic parameters for positioning accuracy improvement. *Robotica*, 9:99–105, 1991.

[14] M. A. Lafortune, P. R. Cavanaugh, H. J. Sommer, and A. Kalenka. Three-dimensional kinematics of the human knee during walking. *Journal of Biomechanics*, 25(4):347–357, Apr. 1992.

[15] J. Lai and H. X. Lan. Identification of dynamic parameters in lagrange robot model. In *Proceedings of 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pages 90–93, 1988.

[16] R. Liscano, H. El-Zorkany, and I. Mufti. Identification of the kinematic parameters of an industrial robot. In *Proceedings of COMPINT '85: Computer Aided Technologies*, pages 477–480, Montreal, Quebec, Canada, 1985. National Research Council of Canada, IEEE Computer Society Press. Held in Montreal, Quebec, Canada, 8–12 September 1985.

[17] B. M. Nigg and W. Herzog, editors. *Biomechanics of the Musculo-skeletal system*. John Wiley, New York, 1998.

[18] M. M. Panjabi, V. K. Goel, and S. D. Walter. Errors in kinematic parameters of a planar joint: Guidelines for optimal experimental design. *Journal of Biomechanics*, 15(7):537–544, 1982.

[19] M. M. Panjabi, V. K. Goel, S. D. Walter, and S. Schick. Errors in the center and angle of rotation of a joint: An experimental study. *Journal of Biomechanical Engineering*, 104:232–237, Aug. 1982.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C, second edition*. Cambridge University Press, 1994.

[21] J. J. Spiegelman and S. L.-Y. Woo. A rigid-body method for finding centers of rotation and angular displacements of planar joint motion. *Journal of Biomechanics*, 20(7):715–721, 1987.

[22] F. C. T. van der Helm. Analysis of the kinematic and dynamic behavior of the shoulder mechanism. *Journal of Biomechanics*, 27(5):527–550, May 1994.

[23] F. C. T. van der Helm. A finite element musculoskeletal model of the shoulder mechanism. *Journal of Biomechanics*, 27(5):551–569, May 1994.

[24] F. E. Veldpaus, H. J. Woltring, and L. J. M. G. Dortmans. A least-squares algorithm for the equiform transformation from spatial marker co-ordinates. *Journal of Biomechanics*, 21(1):45–54, 1988.

[25] D. W. Williams and D. A. Turcic. An inverse kinematic analysis procedure for flexible open-loop mechanisms. *Mechanism and Machine Theory*, 27(6):701–714, 1992.

[26] H. Zhuang and Z. S. Roth. A linear solution to the kinematic parameter identification of robot manipulators. *IEEE: Transactions on Robotics and Automation*, 9(2):174–185, 1993.