Automatic Generation of Kinematic Models for the Conversion of Human Motion Capture Data into Humanoid Robot Motion

Aleš Ude[1], Curtis Man[2], Marcia Riley[3], and Christopher G. Atkeson[3]

[1] ATR International, Information Sciences Division
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
aude@isd.atr.co.jp
[2] ATR Human Information Processing Laboratories, Dept. 3
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
[3] College of Computing, Georgia Institute of Technology
801 Atlantic Drive, GA 30332-0280, USA
mriley@cc.gatech.edu, cga@cc.gatech.edu

Abstract. Human motion capture is a promising technique for the generation of humanoid robot motions. To convert human motion data into the humanoid robot motion, we need to relate the humanoid robot kinematics to the kinematics of a human performer. In this paper we propose an automatic approach for scaling of humanoid robot kinematic parameters to the kinematic parameters of a human performer. The kinematic model is constructed directly from the motion capture data without any manual measurements. We discuss the use of the resulting kinematic model for the generation of humanoid robot motions based on the observed human motions. The results of the proposed technique on real human motion capture data are presented.

1 Introduction

Programming of full-body motions for humanoid robots is a very difficult task because of the complexity of the humanoid robot's kinematics and dynamics. It has recently been demonstrated that human motion capture can be a very powerful technique for the generation of full-body humanoid robot motions such as dance movements [14, 19]. The motion capture approach exploits the similarity between the humanoid robot motion and human motion. It is commonly used in the entertainment industry and computer graphics for the generation of believable animations and is closely related to the idea of imitation learning which has been seen as a means to speed up learning in complex high dimensional motor systems such as humanoid robots [3, 16]. It is not the goal of this paper to present a comprehensive approach to imitation learning for humanoid robots. Here we rather concentrate on a new technique for the automatic generation of synchronized kinematics models of a human performer and humanoid robot. We see this as a first step towards converting human motion into humanoid robot motion, which we consider to be an important step towards higher-level imitation learning.

There exist various commercial systems for human motion capture. They can roughly be classified into three groups: goniometer-based systems, e. g. the Sarcos SenSuit [15], magnetic systems, e. g. the Ascension MotionStar [2], and marker-based optical systems, e. g. the OPTOTRAK from Northern Digital Inc. [12]. Sarcos SenSuit is worn like an exoskeleton and can measure motion directly in the joint space. The main disadvantage of such systems is that they are tailored for a specific robot and come at a specific size, so that only people of proper height can wear them. This work actually started when it turned out that a professional dancer who came to teach our robot an Okinawan folk dance could not put on the SenSuit because she was too short. Magnetic and marker-based optical systems are comparable and are both widely used for human motion capture. Magnetic sensors can measure both position and orientation and can handle occlusions better than optical systems. They are, however, sensitive to metallic objects and noisier than optical systems. In the experiments presented in this work we used the marker-based optical measurement system OPTOTRAK. OPTOTRAK uses identifiable active markers which is advantageous because full-body movements often cause some of the markers to be occluded. Systems using passive markers are more prone to matching errors in such cases.

We also work on a computer vision approach to human motion capture [18]. Vision, being non-invasive, provides a freedom of movement not present with other methods and requires no special preparations such as attaching markers to the body of the performer. However, these approaches are not yet mature enough to enable reliable measurement of complex full-body movements.

1.1 Comparison with Related Research Areas

The automatic construction of kinematic models for robot manipulators is a well established field in robotics [7]. The unknown kinematic parameters are most commonly identified from end-effector pose measurements and robot joint position readings. Unfortunately, such data is not available when constructing a kinematic model of a human. In addition, techniques developed in robotics often require that a robot exercise motion around a single joint with no motion in other degrees of freedom, which is difficult to do for a human. Finally, apart from identifying the kinematic parameters of human motion, we must also determine the positions of markers on the body to enable the future usage of the model by a motion estimation process. This problem does not arise when kinematic models for robot manipulators are generated.

The identification of kinematic models for humans has been studied in the fields of biomechanics and computer graphics. It is well known to the biomechanics community that human joints are not ideal rotatory joints and therefore do not have a fixed center of rotation [8]. Biomechanics researchers often estimate the instantaneous center of rotation instead of fixed joint locations. Our main goal is the generation of humanoid robot motions rather than to model human motion as accurately as possible. The joint axes locations on our humanoid robot are fixed, therefore we also keep rotation axes in the kinematic model of a human performer fixed.

Motion capture techniques are often used for the generation of believable computer animations for virtual humans [9]. Generating humanoid robot motions from motion capture data is different from the generation of computer animations because we need to control a real physical device. Apart from the physical properties of an actor performing the desired motions, we must also take into account the properties of the humanoid robot which is supposed to learn the demonstrated actions. The kinematic model to be developed should not only model human motion well, but should also be scalable to the kinematics of the available robot. This aspect is not considered in the techniques proposed in the field of computer graphics [4, 10, 13, 17], where joint locations on the human body are determined without any additional constraints.

1.2 Overview

In the next section we briefly describe the parameter system that we chose for the kinematic modeling of a humanoid robot's and human motion. Section 3 discusses the selection and synchronization of the humanoid's and human's local coordinate systems. This is necessary for relating human motion to the motion in humanoid robot's joint space. Then we propose an optimization algorithm for fitting the positions of joint axes to the measured human motion and discuss the properties of the developed technique. An algorithm for human motion estimation using the generated models is presented in Section 6. We conclude the paper by presenting some experimental results and with the discussion of future directions.

2 The Kinematic Model

We model human motion as an articulated motion of rigid body parts. The placement of a body in Cartesian space is thus determined by the position and orientation of a global body coordinate system rigidly attached to one of the body parts and by the values of joint angles about body axes. The first decision to be made is the selection of an adequate kinematic parameter convention to model motions generated by such a system. Many different parameter systems were proposed by robot researchers in the past. For example, Zhuang et al. [20] proposed a CPC model as a good choice for the calibration of a robot's kinematic parameters. However, since the basic structure of our robot is known (see Fig. 2) and should be incorporated into the model, we used a model suitable for such a task rather than a parameter system with some kind of general optimal properties for calibration. It turns out that DB's geometric structure can be easily incorporated into the model when using a twist representation [11].

All DB's joints are revolute joints. We model the human body motion by the same type of joints to avoid later conversions between different joint types. Our goal is to determine the location and direction of joint axes on the human body, therefore it is natural to specify the body kinematics using directly the locations and directions of joint axes. Such modeling allows us to easily introduce constraints on joint axes positions and directions. Twists are defined by these parameters. Let $\boldsymbol{n}_i$ be the unit vector in the direction of the revolute joint axis and let $\boldsymbol{q}_i$ be any point on the axis, both given in a global body coordinate system at zero configuration, i. e. with all joint angles θ_i equal to 0. In this case, the twist $\boldsymbol{\xi}_i$ describing the motion about i-th joint axis has the form

$$\boldsymbol{\xi}_i = \begin{bmatrix} -\boldsymbol{n}_i \times \boldsymbol{q}_i \\ \boldsymbol{n}_i \end{bmatrix}, \tag{1}$$

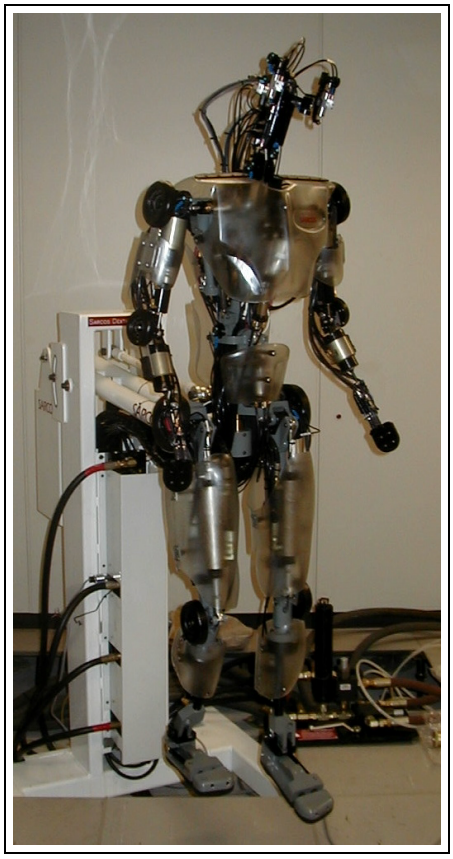

Fig. 1. Humanoid robot DB in our laboratory

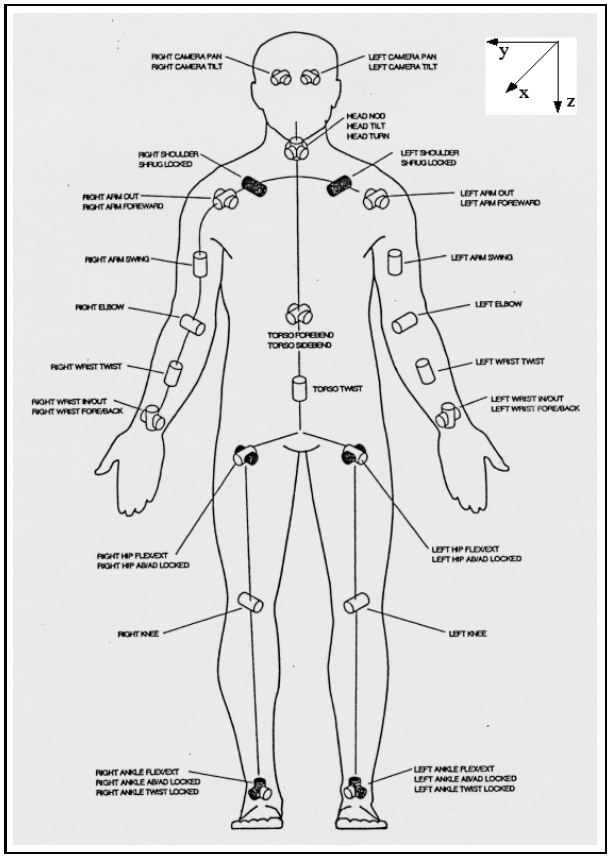

Fig. 2. Kinematic structure of the robot. In the upright position with extended arms and legs, all joint axes are parallel to one of the three main axes of the body (forward/backward: x-axis, left/right: y-axis, up/down: z-axis).

The displacement of a body part caused by motion about such a joint by an angle θ_i can be calculated using the exponential map [11]

$$\exp(\theta_i \boldsymbol{\xi}_i) = \begin{bmatrix} \boldsymbol{R}(\theta_i \boldsymbol{n}_i) & (\boldsymbol{R}(\theta_i \boldsymbol{n}_i) - \boldsymbol{I})(\boldsymbol{n}_i \times (\boldsymbol{n}_i \times \boldsymbol{q}_i)) - \theta_i \boldsymbol{n}_i \boldsymbol{n}_i^T (\boldsymbol{n}_i \times \boldsymbol{q}_i) \\ 0 & 1 \end{bmatrix}, \tag{2}$$

where $\boldsymbol{R}(\theta_i \boldsymbol{n}_i)$ is the orthogonal matrix describing the rotation by θ_i about the unit axis $\boldsymbol{n}_i$.

To generate humanoid robot motion from human motion capture data, we need to relate the human joint motion to the 3-D marker motion. If the coordinates of a marker in a local coordinate system of a rigid body part to which it is attached are given by $\boldsymbol{y}_j$, then its 3-D position at body configuration $(\boldsymbol{r}, \boldsymbol{d}, \theta_1, \ldots, \theta_n)$ can be calculated as follows

$$\tilde{\boldsymbol{y}}_j = \boldsymbol{g}(\boldsymbol{r}, \boldsymbol{d}) \cdot \exp(\theta_{i_1} \boldsymbol{\xi}_{i_1}) \cdot \ldots \cdot \exp(\theta_{i_{n_j}} \boldsymbol{\xi}_{i_{n_j}}) \cdot \boldsymbol{G}_x \cdot \boldsymbol{y}_j. \tag{3}$$

Here $\boldsymbol{\xi}_{i_1}, \ldots, \boldsymbol{\xi}_{i_{n_j}}$ are the twists that describe the kinematic chain generating the motion of the marker $\boldsymbol{y}_j$, $\boldsymbol{G}_x$ is the homogeneous matrix combining the position and orientation of the local body part coordinate system to which the marker is attached with respect to the global body coordinate system at zero configuration (x = torso, pelvis, head, left upper arm, left lower arm, left hand, right upper arm, right lower arm, right hand, left upper leg, left lower leg, left foot, right upper leg, right lower leg, right foot), $\boldsymbol{r}$ and $\boldsymbol{d}$ are the orientation[1] and position of a global body coordinate system with respect to the world coordinate system, and $\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{d})$ denotes the homogeneous matrix corresponding to $\boldsymbol{r}$ and $\boldsymbol{d}$. Note that the set of twists affecting the motion of a marker varies with the identity of the body part to which the marker is attached.

[1] We use rotation vectors to represent the orientation.

In this setting, the kinematic modeling involves the specification of joint axes positions and directions and the placement of local body part coordinate systems, all these parameters being specified in the global body coordinate system when the body is at zero configuration. Apart from local body part coordinate systems, we need to fix also the global body coordinate system in which the position and orientation of the body in space are specified.

3 Choice of Coordinate Systems

We model human motion as an articulated motion of rigid body parts. Such motion can be specified by a number of transformations between local body part coordinate systems. Regardless of the kinematic parameter system in use, the actual values of the parameters depend on the choice of local coordinate systems because they specify transformations between them. Therefore it is essential that local coordinate systems are chosen in such a way that the parameter values at every body posture are the same as the robot's joint angles at the equivalent robot posture.

Our humanoid robot DB (see Fig. 1) has altogether 30 degrees of freedom (see Fig. 2). Its joint angles are all zero when it stands in an upright position with fully extended arms and legs. In addition, in this configuration all the joint axes are parallel to one of the three mutually orthogonal directions along the three main body axes (up/down, left/right, forward/backward). Since it is our goal to transform the captured human motion into DB's motion, the performer's kinematics should be modelled using the same kinematic structure as DB's, only scaled to the physical size of the human performer. This goal will be achieved if we place local coordinate systems on the performer's body parts in such a way that they are all aligned when the performer stands in the upright position with extended arms and legs and that their axes are parallel to the joint axes in this configuration.

The position of a marker is constant only in a local coordinate system of a rigid body part to which it is attached. Hence to estimate human body postures from motion capture data, we need to know also the coordinates of each marker in the local coordinate system of the corresponding body part.

3.1 Practical Determination of Coordinate Frames and Local Marker Coordinates

We ask the performer to stand in an upright position with extended arms and legs. The positions of all markers in this configuration are measured. For each body part, the centroid of all markers attached to it is calculated and the origin of a local body part coordinate system is assumed to coincide with this centroid. Since all local coordinate systems should be aligned in this configuration (with axes parallel to the body axes), the axes of all of them will be fully specified if we can figure out the directions of main body axes in this configuration.

To accomplish this, we exploit the fact that our system knows where up and down is[2]. We can therefore set the z body axis to be parallel to the z-axis of the world coordinate system. The y-axis can be estimated either by attaching two markers at the opposite positions on the left and right shoulder or by estimating the location of shoulder joints from the data. This can be done using one of the methods described in [13, 17]. See Section 5 for the discussion of problems associated with the direct estimation of joint locations. In both cases, the line between the two opposite positions defines the y body axis. Due to inaccuracies in our system, the estimated y- and z-axis are not exactly orthogonal, therefore we enforce the orthogonality by calculating a new z-axis direction

$$\boldsymbol{z} = \left(\frac{\boldsymbol{y}_l - \boldsymbol{y}_r}{\|\boldsymbol{y}_l - \boldsymbol{y}_r\|} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \times \frac{\boldsymbol{y}_l - \boldsymbol{y}_r}{\|\boldsymbol{y}_l - \boldsymbol{y}_r\|}. \tag{4}$$

This new direction is orthogonal to the y-axis and deviates from the original z-axis the least among all directions orthogonal to y. Finally, the x-axis can be calculated using the cross product.

The origin of the global body coordinate system is assumed to be in the middle of a line connecting the two shoulder joints (or two specially placed shoulder markers). It is rigidly attached to the torso. At zero configuration, the global body coordinate system is aligned with the local body part coordinate systems. Using this assumption we can calculate the position and orientation of each body part at zero configuration in the global body coordinate system ($\boldsymbol{G}_x$ in Eq. (3)) and transform the world coordinates of the markers into the local coordinates ($\tilde{\boldsymbol{y}}_j$ and $\boldsymbol{y}_j$ in Eq. (3)).

The rest is simple mathematics. Since the joint angles are equal to 0, Eq. (3) simplifies to

$$\tilde{\boldsymbol{y}}_j = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ 0 & 1 \end{bmatrix} \cdot \boldsymbol{G}_x \cdot \boldsymbol{y}_j. \tag{5}$$

[2] This is done by aligning the z-coordinate of the calibration object with the up/down direction in the real world.

Let $\tilde{\boldsymbol{y}}_l$ and $\tilde{\boldsymbol{y}}_r$ be the coordinates of the two special markers attached to the left and right shoulder (or the locations of the left and right shoulder joint), respectively. The position of a global body coordinate system is given by $\boldsymbol{d} = (\boldsymbol{y}_l + \boldsymbol{y}_r)/2$, while the orientation is given by the rotation matrix $\boldsymbol{R}$ having the three coordinate axes in its columns

$$\boldsymbol{R} = \left[\frac{\boldsymbol{y}_l - \boldsymbol{y}_r}{\|\boldsymbol{y}_l - \boldsymbol{y}_r\|} \times \boldsymbol{z},\ \frac{\boldsymbol{y}_l - \boldsymbol{y}_r}{\|\boldsymbol{y}_l - \boldsymbol{y}_r\|},\ \boldsymbol{z} \right] \tag{6}$$

The orientation of each local body part coordinate system at zero configuration is the same as the orientation of the global body coordinate system. Thus the relative orientation is given by the identity matrix. The relative position can be calculated by subtracting the position of the global body coordinate system from the position of local coordinate systems. Thus for the body part x we have

$$\boldsymbol{G}_x = \begin{bmatrix} \boldsymbol{I}, & \frac{1}{n_x} \sum\limits_{\forall j \in x} \tilde{\boldsymbol{y}}_j - \boldsymbol{d} \\ 0, & 1 \end{bmatrix}, \tag{7}$$

where n_x is the number of markers attached to the body part. Finally, the local coordinates of markers $\boldsymbol{y}_j$ can be obtained by inverting Eq. (5).

4 Estimation of Joint Positions

At this point, the joint axes locations are the only parameters that still need to be estimated. To estimate these parameters, the subject is asked to perform a set of movements which are measured by a motion capture system. He or she should exercise motions around all relevant degrees of freedom if the method is to return an unambiguous answer. Instead of trying to estimate all joint locations in one big optimization process, we decided to split the estimation in six separate optimization problems, one for each of the six kinematic chains whose joint axes locations are to be estimated: head, torso, left and right arm and left and right leg.

The position of a joint axis in 3-D space has only two degrees of freedom because the position along the axis is arbitrary. Recall from Section 3 that directions of all joint axes at zero configuration are parallel to the coordinate axes of the global body coordinate system and are thus all given by one of the vectors $[1,0,0]^T$, $[0,1,0]^T$, $[0,0,1]^T$, $[-1,0,0]^T$, $[0,-1,0]^T$ and $[0,0,-1]^T$. It follows that we can parameterize the location of each joint axis by two coordinates different from the non-zero coordinate of the direction vector of the joint axis. It is easy to see that the coordinate corresponding to the non-zero coordinate of the joint axis direction vector does not influence the resulting twist given by Eq. (1). For example, for a joint axis located at $[a,b,c]^T$ and parallel to $[0,1,0]^T$, the corresponding twist is equal to

$$\boldsymbol{\xi} = \begin{bmatrix} -\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -c \\ 0 \\ a \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{8}$$

and is thus independent of the second coordinate, i. e. b.

Now that we have the independent coordinates for the joint axes locations, we can estimate them by minimizing a suitable optimization criterion. Apart from joint axes locations, we also need to estimate the position and orientation of the body in space as well as the joint angles to match the model markers with the measured marker positions. To make the optimization process smaller, we estimate the body positions and orientations during motion separately from other parameters using the method of Arun et al. [1]. The optimization process then involves only the joint angle locations and joint angles. For instance, to estimate the joint locations affecting the head motion, we need to minimize the following optimization criterion:

$$f(a,b,c,d,e,f,\theta_1(t_k),\theta_2(t_k),\theta_3(t_k)) = \sum_{k=1}^{N} \sum_{j \in \text{head}} \| \boldsymbol{g}(\boldsymbol{r}(t_k),\boldsymbol{d}(t_k)) \cdot \exp(\theta_1(t_k)\boldsymbol{\xi}_1(a,b)) \cdot \exp(\theta_2(t_k)\boldsymbol{\xi}_2(c,d)) \cdot \exp(\theta_3(t_k)\boldsymbol{\xi}_3(e,f)) \cdot \boldsymbol{G}_{\text{head}} \cdot \boldsymbol{y}_j - \tilde{\boldsymbol{y}}_j(t_k) \|^2.$$

Note that the number of parameters increases with the number of measurement times. We applied the trust region method implemented in the MATLAB Optimization Toolbox [5] to minimize this criterion. This method requires the calculation of first derivatives of criterion function f. The calculation of these derivatives is based on the calculation of the Jacobian of the forward kinematics function (3) with respect to joint angles $\boldsymbol{\theta}_1(t_k)$, $\boldsymbol{\theta}_2(t_k)$, $\boldsymbol{\theta}_3(t_k)$, $k = 1, \ldots, N$, and joint locations parameters a, b, c, d, e, f. The calculation of the Jacobian with respect to the joint angles is a classic problem and we implemented it using the approach described in [11]. The calculation of the kinematics Jacobian with respect to the joint axes location parameters is not much different. We omit the details here. The resulting Jacobian is sparse and we use the sparse matrix algebra to reduce the computation time.

In the above criterion function we did not assume that joint axes at joints with multiple degrees of freedom intersect. It would be easy to incorporate such an assumption in our optimization criterion function. We have not done this because joint axes at joints with multiple degrees of freedom do not intersect both in the case of real-world human motion as well as in DB's kinematic model. It would also be possible to simultaneously estimate some of the joint axes locations, e. g.left and right shoulder axes locations, and to impose symmetry constraints in the optimization criterion.

5 Discussion of the Technique

A disadvantage of the above approach is that a human performer cannot move exactly to the zero configuration. This makes our mapping from the person's kinematic parameters to DB's joint angles somewhat inaccurate. Ideally, one would like to find the zero configuration and place the local body part coordinate systems without asking the performer to move to the zero configuration.

One could start by placing local coordinate frames at arbitrary positions and orientations and then use the method of O'Brien et al. [13] or Silaghi et al. [17] to estimate the location of each joint in the local coordinate systems of body parts that are connected by it. Similarly as the method presented in this paper, these two approaches require the user to perform a set of motions that exercise all degrees of freedom of the joints. Both approaches are based on the estimation of positions and orientations of all body parts during motion. These poses can be estimated if for each body part at least three markers attached to it are visible at each measurement time (using for instance the method of Arun et al. [1]). Under certain assumptions about the geometry of the human body, one could then scale the generic kinematic model from Fig. 2 to the actual measurements of the performer using the relative distances between the joints.

While we could adapt the twists $\boldsymbol{\xi}_i$ and the relative poses $\boldsymbol{G}_x$ from Eq. (3) to the size of the performer in this way, the problem of estimating the coordinates of the markers in the local body part coordinate systems ($\boldsymbol{y}_j$ in Eq. (3)) still remains to be solved because the arbitrary selected local coordinate frames are not aligned with the local coordinate frames in the kinematic model of the robot. It might be possible to use the estimated joint locations and to make some assumptions about the geometry of the body to estimate this data, but there is no clear systematic way to do it. In addition, the estimation of joint locations is a very noisy process and the estimates vary depending on the movements that were actually performed. Moreover, the joints of a human are not perfect rotary joints and are not fixed during motion. Thus there is no guarantee that the model parameters generated in this way would be more accurate than the model generated by the proposed approach. For these reasons we decided to stick with our original approach which is simpler to implement.

The model of a human performer does not need to be estimated from scratch at the beginning of every motion capture session. In the next motion capture session, we only need to measure the position of markers at the zero configuration. Using the approach from Section 3, we can then estimate the new local marker positions and the positions and orientations of local coordinate frames in the global body coordinate system. Provided that the two special shoulder markers are always placed on the same two spots, the joint axes positions remain the same as in the old kinematic model and can be used again without performing a repertoire of motions in order to estimate them anew.

6 Motion Estimation from Marker Data

Our trajectory planning method should generate motions which are perceptually similar to the motion of the performer. To attain this, we should minimize the difference between the measured marker positions and the marker positions generated by the recovered joint angles for each frame of motion over the set of body configurations $(\boldsymbol{r}(t_k), \boldsymbol{d}(t_k), \theta_i(t_k))$ [14]

$$\epsilon = \sum_{j=1}^{n} \| \boldsymbol{g}(\boldsymbol{r}(t_k), \boldsymbol{d}(t_k)) \cdot \exp(\theta_{i_1}(t_k)\boldsymbol{\xi}_{i_1}) \cdot \ldots \cdot \exp(\theta_{i_{n_j}}(t_k)\boldsymbol{\xi}_{i_{n_j}}) \cdot \boldsymbol{G}_x \cdot \boldsymbol{y}_j - \tilde{\boldsymbol{y}}_j(t_k) \|^2$$

Fig. 3. Human performers

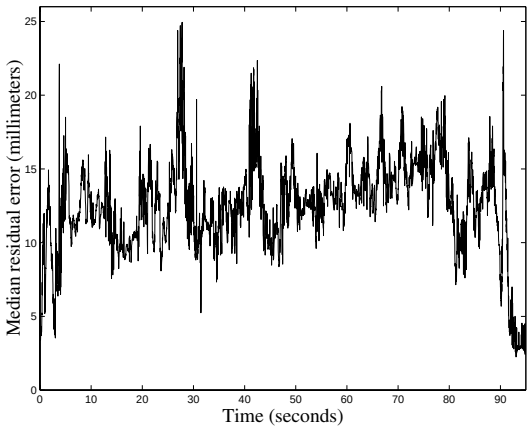

Fig. 4. The median of the difference between the measured marker positions and the marker positions generated by the recovered joint angles at each measurement time

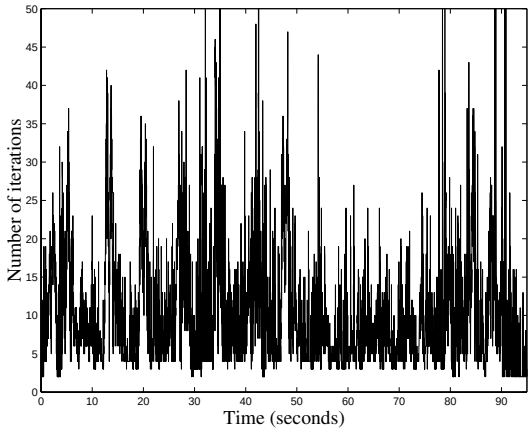

Fig. 5. The number of iterations needed by *lsqnonlin* to converge

$$= \sum_{j=1}^{n} \|\boldsymbol{h}_j(\boldsymbol{r}(t_k), \boldsymbol{d}(t_k), \theta_1(t_k), \ldots, \theta_n(t_k)) - \tilde{\boldsymbol{y}}_j(t_k)\|^2$$
$$= \|\boldsymbol{h}(\boldsymbol{r}(t_k), \boldsymbol{d}(t_k), \theta_1(t_k), \ldots, \theta_n(t_k)) - \tilde{\boldsymbol{y}}(t_k)\|^2, \quad (9)$$

where $\boldsymbol{h} = [\boldsymbol{h}_1^T, \ldots, \boldsymbol{h}_n^T]^T$ and $\boldsymbol{y}(t_k) = [\boldsymbol{y}_1(t_k)^T, \ldots, \boldsymbol{y}_n(t_k)^T]^T$. This is a standard nonlinear least-squares optimization problem and there are several good methods that can be applied to solve it [5].

Experiments have shown that the incorporation of constraints into the above optimization criterion often increases the quality of the reconstructed motion. For example, the elbow joint angle is equal to 0 when the performer fully extends his arm. It is visually very disturbing if the estimated elbow angle turns out to be negative because of the measurement noise and/or inaccuracies in the kinematic model. Thus it is sensible to add simple bounds, which limit the angles to the area that can actually be reached by a human, to the above optimization criterion

$$l_i \leq \theta_i \leq u_i. \quad (10)$$

Efficient algorithms are available for the minimization of the nonlinear least squares criterion (9) subject to box constraints (10). We utilized the MATLAB Optimization Toolbox function *lsqnonlin*, which requires the calculation of the kinematics Jacobian, to solve such optimization problems.

Good initial approximations for the body posture are available when estimating human motion over time (we simply take the previous estimate as an initial estimate at the next time step) and *lsqnonlin* finds an optimal solution in only a few iteration steps (see also Section 7). Such an initial approximation is not available at the beginning of motion. A different approach should therefore be used in the first data frame. If at least three markers are visible on a body part, then we can estimate its position and orientation in space. The joint angle(s) between two connected

Joint	x-axis location			y-axis location			z-axis location		
Waist	0	0.234	319.0	-61.05	0	381.47	-61.05	0.234	0
Neck	0	7.1936	-75.43	-53.28	0	-101.0	-45.14	2.0611	0
Left shoulder	0	-156.7	29.68	-97.74	0	6.0574	-97.74	-156.7	0
Left elbow				-122.9	0	309.62			
Right shoulder	0	143.73	22.68	-83.02	0	4.1474	-83.02	143.73	0
Right elbow				-99.27	0	315.68			
Left hip	0	-96.18	429.8	-89.79	0	450.51			
Left knee				-18.97	0	923.69			
Left ankle				-119.3	0	1344.5			
Right hip	0	89.066	448.9	-47.13	0	451.17			
Right knee				-50.96	0	931.15			
Right ankle				-119.3	0	1344.5			

Table 1. The estimated locations of joint axes (in millimeters). Note that the symmetry constraint was enforced when calculating the left and right ankle axis locations.

Fig. 6. Computer animation of the captured data

body parts can then be estimated using an analytic inverse kinematics method. If less than three markers attached to one of the two connected body parts are visible, then this method cannot be used. In this case the corresponding joint angles are set to default values. We were able to achieve good convergence properties with such initialization.

DB's joint limits are stricter than human joint limits. It is possible to replace the human joint limits with DB's joint limits in (10). This approach trades off joint errors and Cartesian target errors in a straightforward way and makes the robot see only feasible postures when interpreting or reasoning about the captured motion.

Another possibility for making the optimization process more reliable is to recover a complete trajectory instead of single configurations. Such an approach opens many additional possibilities which we explored elsewhere [19].

7 Results

Although extensive experiments to test the precision of the generated kinematic models are yet to be carried out, we can say that initial experiments have shown that our model construction process is accurate enough for the generation of both computer animations and humanoid robot motions. The values of an automatically constructed kinematic model are shown in Table 1. They are consistent with the manually measured limb lengths of the left performer in Fig. 3.

Some frames from a computer animation and from a humanoid robot motion that were generated based on the approaches presented in this paper are shown in Fig. 6 and 7. In the karate experiment, we were able to estimate

karate kata movements which were over one and a half minutes long without the motion estimation process ever to break down. In the Okinawa dance experiment, the motion capture sessions were shorter than ten seconds and longer motions were generated by concatenation (see also [14]). A PD controller running at 420 Hz was used for trajectory tracking and the temporal aspects of motion were maintained. We are also experimenting with a better controller based on the inverse dynamics of DB. Fig. 4 illustrates that the marker positions generated by the recovered joint angles (used to generate computer animations in Fig. 6) follow the measured markers well. This shows that the rather limited kinematic structure of DB is sufficient to reproduce quite complex human motions. Finally, Fig. 5 depicts the number of iterations needed by *lsqnonlin* to converge. The median number of iterations for this sequence was 7 and the average number of iterations was 10.

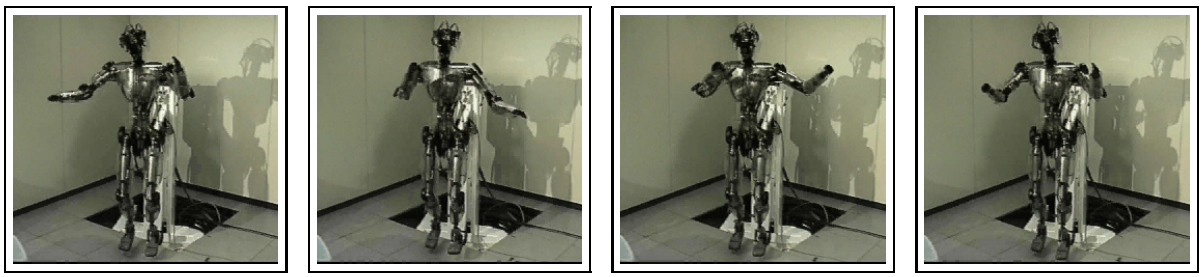

Fig. 7. Humanoid robot performing Okinawan dance movements

8 Conclusions

We have presented an automatic technique for the generation and synchronization of kinematic models describing human and humanoid robot motion. Our technique involves measuring marker positions at the zero configuration and over a repertoire of motions exercising all relevant degrees of freedom. No manual measurement of the performer's limb lengths is necessary. The structure of the generated kinematic model is kept the same as the kinematic structure of the humanoid robot under consideration, but the joint axes positions are scaled to the lengths of the human performer. Using the approaches presented in this paper, we were able to process different motion sequences and to generate both computer animations and humanoid robot motions.

The difference between the motion capabilities of a human performer and of a humanoid robot is a serious problem when converting the motion capture data. Constraining the joint limits in the optimization criterion does result in feasible motions, but the "style" or "essence" of motion might suffer. In the future we shall work on the development of algorithms that identify what is important to preserve in the captured motion, and what is less important or irrelevant and can thus be omitted. Another problem that should be addressed in the future is the difference in physical characteristics, such as size and mass, between DB and the human performer. It was shown in the field of computer graphics that control algorithms must be adapted when simulating motion of characters having different physical properties. This problem was not so severe in our present experimental setting because the robot is mounted at the pelvis so that we do not have to worry about balance. However, the algorithms for geometric and mass scaling [6] might be very useful for the transfer of human motion to the free standing humanoids.

Acknowledgment: Aleš Ude is on leave from the Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia.

References

1. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(5):698–700, September 1987.
2. Ascension Technology Corporation. http://www.ascension-tech.com/.
3. C. G. Atkeson, J. Hale, M. Kawato, S. Kotosaka, F. Pollick, M. Riley, S. Schaal, T. Shibata, G. Tevatia, A. Ude, and S. Vijayakumar. Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, to appear.
4. B. Bodenheimer and C. Rose. The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97, Proc. Eurographics Workshop*, pages 3–18. Springer-Verlag, 1997.

5. T. Coleman, M. A. Branch, and A. Grace. *Optimization Toolbox User's Guide*. The MathWorks, Natick, MA, 1999.
6. J. K. Hodgins and N. S. Pollard. Adapting simulated behaviors for new characters. In *Computer Graphics, Proceedings of SIGGRAPH '97*, pages 153–162, August 1997.
7. B. Karan and M. Vukobratović. Calibration and accuracy of manipulation robot models - an overview. *Mechanism and Machine Theory*, 29(3):479–500, 1994.
8. M. A. Lafortune, P. R. Cavanaugh, H. J. Sommer, and A. Kalenka. Three-dimensional kinematics of the human knee during walking. *J. Biomechanics*, 25(4):347–357, April 1992.
9. R. Maiocchi. 3-D character animation using motion capture. In N. Magnenat Thalmann and D. Thalmann, editors, *Interactive Computer Animation*, pages 10–39. Prentice-Hall, London, 1996.
10. T. Molet, R. Boulic, and D. Thalmann. A real time anatomical converter for human motion capture data. In R. Boulic and T. Hégron, editors, *Computer Animation and Simulation '96, Proc. Eurographics Workshop*, pages 79–94. Springer-Verlag, Vienna, 1996.
11. R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, New York, 1994.
12. Northern Digital Inc.
http://www.ndigital.com/.
13. J. F. O'Brien, R. E. Bodenheimer, G. J. Brostow, and J. K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proc. Graphics Interface 2000 Conference*, Montreal, Canada, May 2000.
14. M. Riley, A. Ude, and C. G. Atkeson. Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching. In *Proc. 2000 Workshop on Interactive Robotics and Entertainment*, pages 35–42, Pittsburgh, Pennsylvania, April/May 2000.
15. Sarcos.
http://www.sarcos.com/.
16. S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.
17. M.-C. Silaghi, R. Plänkers, R. Boulic, P. Fua, and D. Thalmann. Local and global skeleton fitting techniques for optical motion capture. In N. Magnenat Thalmann and D. Thalmann, editors, *Modeling and Motion Capture Techniques for Virtual Environments*, pages 26–40. Lecture Notes in Computer Science 1537. Springer-Verlag, Berlin, New York, 1998.
18. A. Ude. Robust estimation of human body kinematics from video. In *Proc. IEEE/RSJ Conf. Intelligent Robots and Systems*, pages 1489–1494, Kyongju, Korea, October 1999.
19. A. Ude, C. G. Atkeson, and M. Riley. Planning of joint trajectories for humanoid robots using B-spline wavelets. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2223–2228, San Francisco, California, April 2000.
20. H. Zhuang and Z. S. Roth. A linear solution to the kinematic parameter identification of robot manipulators. *IEEE Trans. Robotics Automat.*, 9(2):174–185, April 1993.