Available online at www.sciencedirect.com
SCIENCE DIRECT®
Image and Vision Computing 22 (2004) 843–850
image AND vision COMPUTING
www.elsevier.com/locate/imavis

A procedure for automatically estimating model parameters in optical motion capture

Maurice Ringer*, Joan Lasenby

Department of Engineering, Cambridge University, Cambridge CB2 1PZ, UK

Received 12 March 2003; received in revised form 17 November 2003; accepted 12 February 2004

Abstract

Model-based optical motion capture systems require knowledge of the position of the markers relative to the underlying skeleton, the lengths of the skeleton's limbs, and which limb each marker is attached to. These model parameters are typically assumed and entered into the system manually, although techniques exist for calculating some of them, such as the position of the markers relative to the skeleton's joints.

We present a fully automatic procedure for determining these model parameters. It tracks the 2D positions of the markers on the cameras' image planes and determines which markers lie on each limb before calculating the position of the underlying skeleton. The only assumption is that the skeleton consists of rigid limbs connected with ball joints. The proposed system is demonstrated on a number of real data examples and is shown to calculate good estimates of the model parameters in each.

Keywords: Optical motion capture; Training sequence; Computed tomography

1. Introduction

Modern techniques for optical motion capture involve simultaneously estimating the pose of the underlying skeleton of the subject while tracking the movement of the markers on the cameras' image planes [7,9,15]. Knowledge of how the skeleton moves constrains how the markers move and allows capture of more complex motion than is possible with model-less marker tracking.

Tracking systems based on skeletal models, however, require knowledge of the location of the markers relative to the skeleton, particularly to the joints, as well as the lengths of each of the limbs. This information changes with each subject as new markers are attached and as the size of the subject varies. The location of the joints relative to the markers are of particular interest to doctors, physiotherapists and sports motion analysts who desire it to analyse the precise movements of patients or athletes. Such information could be used, for example, to aid the construction of artificial hips or knee joints for a particular patient. Typically, these model parameters are assumed known and entered into the tracking system manually [11].

This paper presents a technique for calculating these model parameters—which limb each marker is attached to, the location of the markers relative to the underlying skeleton, and the lengths of each limb—given the 2D co-ordinates of the bright points detected at each camera's image plane. The procedure is fully automatic.

Recently, techniques have been proposed to calculate much of this information, particularly the location of the joints [3,5,13,17,18], however, all of these require the 3D trajectory of each marker and knowledge of which limb each marker is attached to. And in order to calculate the 3D trajectory of each marker, it is necessary to track their movement at the camera image plane, which is difficult without a model of the underlying skeleton.

For this reason, we propose that the subject perform a short training sequence prior to performing the motion that is to be captured. The movement during the training sequence should be slower than for typical motion capture and efforts should be made so that every marker can be observed by at least two cameras for most of the sequence. Motions of this nature can be tracked without a model, and the resulting track can be used to calculate the model

* Corresponding author.
E-mail addresses: mar39@eng.cam.ac.uk (M. Ringer), jl@eng.cam.ac.uk (J. Lasenby).

0262-8856/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2004.02.011

parameters, which in turn are used to track more complex motion. Each limb should also be rotated about its joints during the training sequence, so that the joint locations can be identified in the subsequent processing.

The suggestion of a training sequence to calculate model parameters can also be found in Ref. [9], however, their methods are not automatic and use a non-optimal iterative technique for calculating the joint locations. They also assume knowledge of which markers are attached to each limb of the subject.

Our procedure for calculating the model parameters from the training sequence is:

1. Calculate the 3D position of each marker during the training sequence.
2. Calculate which markers are attached to each limb.
3. Calculate the size and position of the underlying skeleton and the location of the markers relative to it.

Sections 2–4 describe these three steps in more detail. Section 5 then details the results of the system when applied to a number of real data examples.

2. Generating 3D marker positions

The first stage in estimating the model parameters is calculating the 3D trajectory of each marker during the training sequence. Let the 3D position of each marker n at time k be $y_n(k)$, where $n \in \{1,\ldots,N\}$ and $k \in \{1,\ldots,K\}$. N is the number of markers and K is the number of frames. The input to this stage is the 2D co-ordinates of the bright points detected on each camera's image plane.

The difficulty in calculating y is that the association between markers and detected points is unknown, i.e. no information is given as to which marker produced each bright point detection. The problem is made worse by the fact that markers are often occluded in certain poses of the subject and that each camera detects only an unknown subset of the markers.

Estimating the 3D marker positions, $y_n(k)$, at each frame using a global optimisation or batch technique is computationally infeasible. Instead, we propose a standard sequential estimation (tracking) procedure [2] to perform this task. The procedure can be broken into three stages, each of which are repeated for each time frame:

1. Predict the position of each marker at time $k, \hat{y}_n(k)$, by projecting forward the current estimate of the marker's trajectory as given by $y_n(k-1), y_n(k-2)$, etc. Either linear or cubic interpolation was used for this.
2. Calculate the most likely marker-to-detection association at time k using $\hat{y}_n(k)$.
3. Calculate the 3D marker position, $y_n(k)$. If the association of step 2 determined that two or more cameras detected marker $n, y_n(k)$ is found with standard triangulation [10]. If only one camera detected marker $n, y_n(k)$ is a compromise between its predicted position, $\hat{y}_n(k)$, and the 3D ray projected through the detection on the camera's image plane from the camera's origin. If no cameras detected marker $n, y_n(k) = \hat{y}_n(k)$.

As mentioned earlier, the most problematic of these three steps is the second. At each time k, we desire to label each detection in each camera with the marker that generated it. The first stage of this process is to project onto each cameras' image plane the predicted position of each marker, $\hat{y}_n(k)$. The problem is then one of matching the projected 2D marker positions to the detections in each camera, and of matching detections between cameras. Fig. 1 shows an example input to this stage of the tracking process.

Let M be the number of cameras and χ be the $(M+1)$-dimensional matrix representing the association we are attempting to determine. $\chi_{i_0,i_1,\ldots,i_M}$ is 1 if marker i_0 generated detection i_1 in camera 1, detection i_2 in camera 2, and so on. If any of these conditions are not true, $\chi_{i_0,i_1,\ldots,i_M}$ is 0. The discrete space of possible associations is constrained by the fact that each marker can be associated with at most one detection in each camera and that any detection in a given camera can be associated with at most one detection in another. Thus, summing all elements onto any one dimension of χ is at most unity (for example, for any i_1, $\sum_{i_0}\sum_{i_2}\cdots\sum_{i_M}\chi_{i_0,i_1,\ldots,i_M} \leq 1$).

Let $c_{i_0,i_1,\ldots,i_M}$ be the cost of assuming that marker i_0 generated detections $i_1,\ldots,i_M$ in cameras $1,\ldots,M$, respectively. The cost of assuming a particular association, χ, is

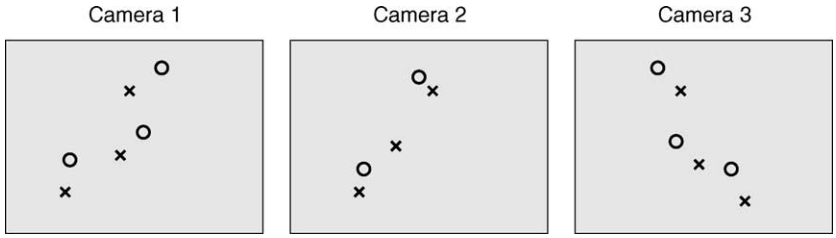

Fig. 1. Example input to the 3D point tracker. The predicted positions of the markers on the camera image planes are shown by the crosses, while the circles mark the detections. The association step requires matching circles to crosses in each camera, and matching circles between cameras. Note that each camera may not detect all markers.

given by,

$$\sum_{i_0=1}^{N}\sum_{i_1=1}^{Q_1}\cdots\sum_{i_M=1}^{Q_M}\chi_{i_0,i_1,\ldots,i_M}c_{i_0,i_1,\ldots,i_M} \tag{1}$$

where Q_m is the number of detections in camera m. In this equation, the non-zero elements of χ select which elements of the cost matrix, c, to include in the summation. We require χ such that Eq. (1) is minimised. That is, we need to select elements of c so that at most one element is taken from each of its dimensions and so that the resulting sum is minimum.

The cost matrix, c, is given by

$$c_{i_0,i_1,\ldots,i_M} = r(i_1,\ldots,i_M) + g\sum_{m=1}^{M} d(i_0,i_m) \tag{2}$$

where $r(i_1,\ldots,i_M)$ is a measure of how well detections $i_1,\ldots,i_M$ reconstruct a single point in space (i.e. how well they satisfy the epipolar constraint [4]), and $d(n,i_m)$ is the L_2 distance between the predicted position of marker n (as projected onto the image plane of camera m) and detection i in camera m.

The constant g in Eq. (2) defines the relative importance between matching predicted marker positions to detections and matching detections between cameras. It is the ratio between the accuracy of the detections (the measurement noise) and the confidence in the predicted marker positions (the process noise).

The problem of searching the constrained space of possible χ for the value which minimises Eq. (1) is often referred to as the $(M+1)$-dimensional linear assignment problem [2].

For the 2D case (for example, associating markers to the detections in a single camera, or the detections between two cameras only), the problem can be considered as selecting elements of the 2D matrix c so that their sum is a minimum and so that at most one element is selected from each row and each column.

For the 2D case, fast and efficient algorithms exist for computing the optimum association [1,12]. However, for higher dimensions ($M \geq 2$) the problem can be shown to be NP-hard.

The general linear data assignment problem has recently received much attention from the radar tracking community who recommend a technique called Lagrangian relaxation for estimating the optimal association [2,14]. Lagrangian relaxation works by first relaxing the uniqueness constraint along all but two of the dimensions so that the problem can be solved, providing a possibly infeasible 'dual' solution. The dual solution is possibly infeasible because it is only guaranteed to sum to at most unity along the 2D chosen for the problem.

From the dual solution, a feasible 'primal' solution can be attained by considering the 2D for which the uniqueness constraint holds as a single dimension and repeating the process. It is not guaranteed that the primal solution is the optimum association, however, the distance between its total cost and the total cost of the dual solution provides a measure of how close it is to the optimum. The solution is refined by weighting the elements of $c_{i_0,i_1,\ldots,i_M}$ that violate the uniqueness constraint in the dual solution by some amount (the Lagrangian multipliers) and calculating new dual and primal solutions.

If the dual solution is ever feasible, it is the optimal solution, however, Lagrangian relaxation is not guaranteed to converge to it. Typically, if the optimal solution has not been reached after some period of time, the process is stopped and the best primal solution (that which was closest to its corresponding dual solution) is retained.

For further information on Lagrangian relaxation, the reader is referred to [2,14].

Using this technique to estimate the most likely marker-to-detection association at each time k in conjunction with the other elements of the 3D point tracker as listed above, we are able to calculate the 3D position of each marker at every time frame but the first. For time $k=1$, no prediction for $y_n(k)$ exists so we instead calculate only the M-dimensional association, $\chi_{i_1,\ldots,i_M}$, between the detections on each camera image plane. It is from this association that we reconstruct $y_n(1)$. Thus, it is necessary for each marker to be visible by at least two cameras in the first frame of the test sequence.

Tracking the markers in this manner is prone to errors if the markers move too far from their predicted positions between time frames and if markers are occluded from all cameras for too long. It is for this reason that we suggest a separate training sequence during which the subject's motions are slow and such that every marker is visible to at least two cameras most of the time.

To the knowledge of the authors, this work provides the first attempt to apply the Lagrangian relaxation technique to computer vision and to solve the problem of matching points between more than two images using an efficient search algorithm over the entire association space.

3. Calculating likely marker-to-limb associations

Given the 3D trajectory of each marker, it is then necessary to determine which markers are attached to which limbs of the subject. Let Ω be the $N \times L$ matrix representing this marker-to-limb association, where L is the number of limbs. $\Omega_{n,l}$ is 1 if marker n is attached to limb l and 0 otherwise (limb l identifies a specific limb on the subject, such as the forearm, or upper leg). We assume that a given marker can lie only on a single limb, yet a given limb may contain any number of markers, thus, any row of Ω may contain only one non-zero entry.

The quality of choice of Ω is given by the algorithms of Section 4, which estimate not only the model parameters,

but also how well the model parameters fit the observed marker trajectories given each association.

The problem of estimating the best value of Ω is, in some sense, similar to that of estimating χ during the tracking process of Section 2—we desire to minimise a cost function over a matrix variable whose elements are {0,1} and whose form is very constrained. Unfortunately, the form of the constraints of Ω (the fact that a single column may contain any number of non-zero entries) and the fact that the cost function is non-linear mean that the Lagrangian relaxation technique described in Section 2 cannot be used to estimate Ω.

We propose, instead, to generate a values of Ω, which are calculated using some knowledge of how markers attached to the same limb are expected to move. That is, we sample the constrained space of possible Ω about a point that we expect to be a good estimate of the best marker-to-limb association. The a proposed marker-to-limb associations are then passed to the algorithms of Section 4 and the true association is determined as the one which provides the best estimate of the model parameters.

From a Bayesian perspective, we seek the maximum of the posterior density function,

$$P(R, e, t, \Omega|y) = P(R, e, t|\Omega, y)P(\Omega|y) \tag{3}$$

where R, e and t are the desired model parameters. The techniques of this section ideally estimate the values of Ω which provide the a maximum values of the second term, $P(\Omega|y)$, while the techniques of Section 4 maximise the first term, $P(R, e, t|\Omega, y)$. Although it is not guaranteed that the most likely association will maximise the entire posterior, is it expected that one of the best a will.

Calculating the a most likely values of Ω is performed in two stages, as illustrated in Fig. 2. First, markers that are attached to the same limb are identified. This is done by observing which markers remain a fixed distance from each other over the duration of the training sequence. We create an $N \times N$ matrix D, where $D_{n1,n2}$ is the variance of the L_3 distance between $y_{n1}(k)$ and $y_{n2}(k)$ over all k. Note that D is symmetric and the elements along its major diagonal are zero. Our procedure for grouping markers on the same limb then proceeds as follows:

1. Determine the minimum element of D. Let the indices of this element be $n1$ and $n2$.
2. If neither $n1$ nor $n2$ has been assigned to a limb, assume a new limb exists and associate these two markers with it.
3. If either $n1$ or $n2$ has been assigned to a limb but not the other, associate the unassigned marker to the limb associated with the other marker.
4. If both markers $n1$ and $n2$ are already assigned to limbs, an inconsistency has occurred. In this case, make no marker-to-limb associations but take note of this marker pair.
5. Set element $D_{n1,n2}$ to a number sufficiently large that it will not be selected again in step 1.
6. Return to step 1 and repeat until all elements of D have been assessed.

This procedure creates a single set of groups of markers, or marker-to-group association. To generate further possible associations, we force the marker pairs that were considered as an inconsistency (in step 4) to be on the same limb. Let a_1 be the total number of marker-to-group associations formed, which is a function of the number of markers, limbs, and inconsistencies found when analysing D.

The second stage of calculating Ω involves associating the groups of markers to a specific limb of the body, such as the forearm, or upper leg. This is done by observing which groups of markers remain close to each other over the duration of the training sequence. We construct a $L \times L$ matrix D' where $D'_{l1,l2}$ is the mean L_3 distance between the centroid (mean) of the groups of markers $l1$ and $l2$ over all k. Like D, D' is also symmetric and contain zeros along its major diagonal. We iteratively select the smallest element from D' and match pairs of marker groups in exactly the same manner as was performed on D. The selected pairs of marker groups are deemed to have a common joint.

The search space for the best group-to-limb associations is lessened by considering the position in 3D space of the centroid of each group in a given frame. For example, if it is known that the subject stands in a neutral pose in the first frame of the training sequence, the groups of markers in the lower half of the field of view need not be considered as possibly attached to the arms or upper body. Elements of D' corresponding to pairs of marker groups that are less likely to be attached for this reason are increased before the search for associations begins.

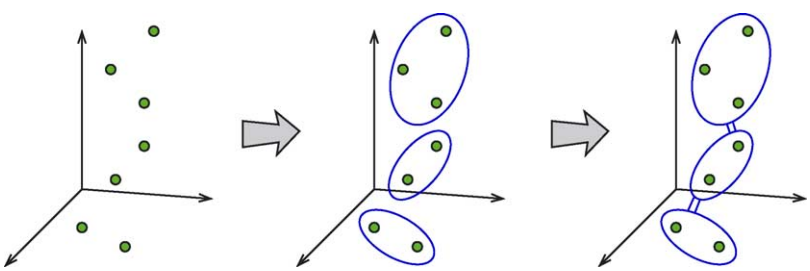

Fig. 2. Assigning markers to limbs is a two step process. From the 3D marker positions (left image), markers attached to the same limb are identified (centre image). Next, groups of markers sharing have a common joint are determined (right image).

Once again, inconsistencies that occur when analysing D' mean that a_2 possible association of groups-to-limbs are generated. This second stage of associating markers to limbs is performed on each of the a_1 marker-to-group associations formed in the first stage, resulting in $a = a_1 a_2$ possible values of Ω for input to the algorithms of Section 4.

4. Calculating the marker offset vectors

Given the 3D position of each marker over time and a knowledge of which limb each marker is attached to, it is possible to calculate the underlying axes of rotations, the 3D positions of the joints around which the rotations occur and the position of the markers with respect to these joints [3,5,13,17,18]. We adopt a similar approach to that of Ref. [5], as the solutions are closed-form and thus quick to calculate. Also, as is shown in Section 5, their techniques usually provide a more accurate estimate of the joint centres.

We initially consider each joint and the pair of limbs rotating about it independently. The general motion of this joint and two limbs is described by

$$R_l(k)e_l^p + t_l(k) = z_l^p(k) \tag{4}$$

where e_l^p is the position of marker p on limb l with respect to the joint, $t_l(k)$ is the trajectory of the joint, $R_l(k)$ be the rotation which defines the orientation of limb l at time k during the training sequence, and $z_l^p(k)$ is the 3D position of marker p on limb l at time k. $z_l^p(k)$ is $y_n(k)$ where n is a function of Ω (n is the row index containing the pth non-zero element in column l). Fig. 3 illustrates this motion of the joint and two limbs.

We first solve for $R_l(k)$ by subtracting different instances of Eq. (4) (for different p and k) and eliminating t and e. Next, we solve for e_l^p by eliminating only t and combining

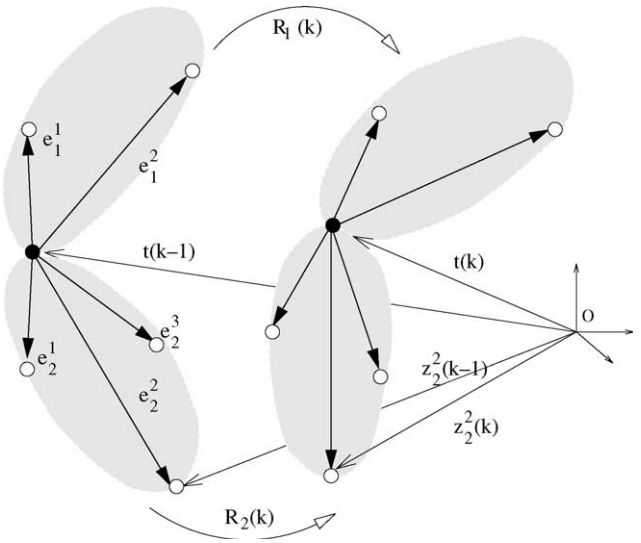

Fig. 3. The motion of two limbs connected by a single ball joint. e is the marker positions relative to the joint, z is the observed position of the markers, R is the rotation of the limbs and t is the position of the joint.

equations of different l, and finally t is determined by substituting the estimates for R and e back into Eq. (4). In each case, the resulting equations are linear and contain excess, but noisy information, so can be solved directly using a least squares method. When estimating the rotation matrices, $R_l(k)$, the method of Ref. [10] is used.

The reader is referred to [5] for more detailed information on estimating e, R and t from y and Ω in this manner.

The above procedure is performed for each of the a marker-to-limb associations calculated in Section 3. The likelihood of e, R and t is a function of the reconstruction error,

$$E = \sum_k \sum_l \sum_p |R_l(k)e_l^p + t_l(k) - z_l^p(k)|^2. \tag{5}$$

We select the marker-to-limb association, Ω, and its corresponding model parameters, e, R and t for which the reconstruction error is smallest.

When constructing a tracker for general motion capture of the subject, we are interested in e, the marker positions relative to the joints, and the length of each limb. The limb lengths are calculated by averaging the L_3 distance between the trajectories of each joint, $t_l(k)$.

5. Results

The system proposed in the previous sections was implemented and tested on a number of real data examples. In each case, a simple training sequence was performed which consisted of moving the limbs of interest slowly and ensuring that rotation occurred at each joint.

In the first test, six markers were attached to two limbs of the subject's arm. For this sequence, the Lagrangian relaxation technique of Section 2 always converged on the correct marker-to-detection association and the 3D point tracker successfully generated trajectories of the markers, $y_n(k)$. The algorithms of Section 3 determined no inconsistencies or doubtful marker-to-limb assignments when analysing the D and D' matrices, and provided only a single, correct, marker-to-limb association, which is shown in Fig. 4. Fig. 4 shows the result of calculating the marker positions relative to the joint, e, and the joint position, t, for this test case.

The second test case involved tracking two arms, wearing 15 markers and modelled as seven limbs (the axis between the shoulders was considered as a limb). Once again, the Lagrangian relaxation technique provided the correct marker-to-detection association and the 3D trajectories of the markers. The algorithms of Section 3 were again certain about which markers were attached to the same limbs, although provided $a_2 = 3$ different group-to-limb suggestions for the model parameter estimation.

Fig. 5 shows the result of the final stage of the model parameter estimation. As can be seen, the location of

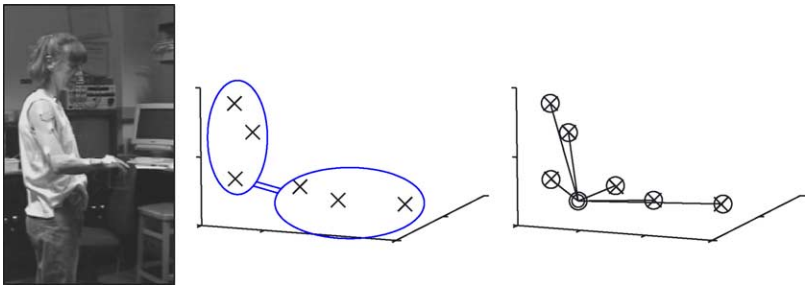

Fig. 4. Result of estimating which markers are attached to each limb (middle image) and the model parameters e (the circles and lines) and $t(k)$ (double circle), for a two limb case. Crosses represent the observed marker positions, $y(k)$.

the underlying skeleton and the position of the markers relative to it, appears to follow the motion of the subject well.

Each training sequence was 20 s and 500 frames long, and calculating the model parameters took about 5 s on a 1 GHz pentium PC.

The third test case for proposed algorithms was provided from the output of a commercial motion capture system. In this case, the 3D trajectories of the markers were provided, so the first stage of parameter estimation process was not required. Also in this case, the subject wore markers on his right thigh and pelvis and, after performing the training sequence and while still wearing the markers, was subjected to a computed tomography (CT) scan from which the exact location of the hip joint centre relative to the markers was determined.

The 3D trajectory of the hip centre was calculated using the procedure described in Section 4, which comes from Gamage et al. [5]. The hip centre trajectory was also calculated using techniques described by Davis et al. [3]

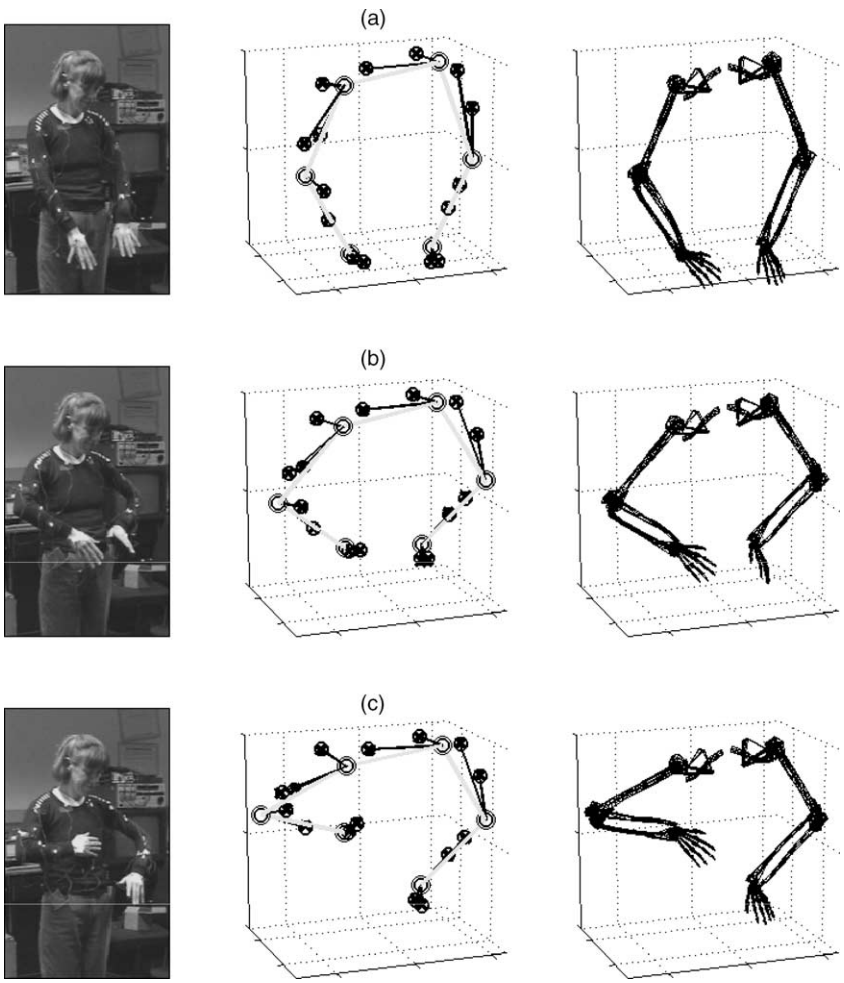

Fig. 5. The calculated underlying skeleton for a seven limb model, shown at three different time frames during the training sequence. The gray lines represent the estimated limbs, the circles and lines represent the marker offset vectors, and the crosses represent the observed marker positions, $y(k)$.

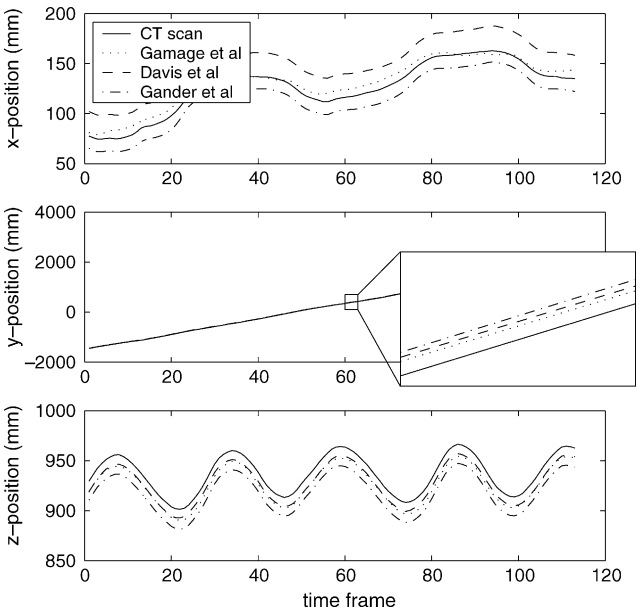

Fig. 6. 3D location of hip joint centre as given by the CT scan and by three estimation techniques, for the third test case.

and by Gander [6]. Davis assumes a specific skeletal shape and geometry, while Gander describes a general sphere-fitting technique. Both techniques are often used by commercial motion capture companies.

Fig. 6 shows the 3D trajectory of the joint as given by these three algorithms compared to the true trajectory as given by the CT scan. The mean squared error between the CT scan path and the three algorithms are:

Gamage et al. 14.3 mm
Davis et al. 27.1 mm
Gander et al. 24.7 mm

As can be seen, our procedure best follows the path of the joint centre.

6. Discussions and conclusions

From the initial tests performed so far, it appears that the proposed procedure for estimating the model parameters works very well. The system calculates which limb each marker is attached to, the location of the markers relative to the underlying skeleton, and the lengths of each limb. It is fully automatic and requires little time to execute.

Of the three stages of the process, it is the first that appears most prone to failure. If the value of χ estimated at any time k does not reflect the true association between detections and markers at that time, the ability to track the 3D trajectories of the markers is lost. This occurs when markers are occluded for long periods of time and the predicted positions of these markers, $\hat{y}_n(k)$, are incorrect. When the predicted positions of the markers are incorrect, so are the elements of the cost matrix, c, and the value of χ that minimises Eq. (1) may not be the true marker-to-detection association.

One method to perhaps overcome these problems is to implement a multiple hypothesis tracker [2,16]. In this system, a modification to the Lagrangian relaxation method would be used to return not the sole best, but the best N estimates of χ which minimise Eq. (1). Propagating each of these estimates through the point tracker creates a trellis of possible associations and marker trajectories, which needs to be pruned to stop from growing exponentially. The cost of a particular path through this trellis could be given by the sum of the association costs at each of the nodes in the path (Eq. (1)).

In this manner, it is not necessary that the true association be the one which provides the best cost at time k, so long as it is one of the best N, thus allowing for more error in the predicted marker positions, $\hat{y}_n(k)$.

Similar to the problem of associating markers to detections is the problem of associating markers to limbs. As mentioned in Section 3, the form of Ω means that fast and efficient techniques such as Lagrangian relaxation cannot be used. The proposed technique of testing a estimates of Ω found by heuristic methods has been found to work well, although a number of other algorithms have been considered and tested.

An alternative approach for optimising over awkward spaces that is currently proving popular in the scientific literature are Markov Chain Monte Carlo (MCMC) methods such as the Metropolis–Hastings algorithm [8,19]. These techniques work by proposing a new sample estimate from a previous one, then accepting or rejecting it depending on its cost. This process is repeated, allowing the sample to move about the space of possible solutions, and after some time the sample with the best cost is selected. In the case of selecting the best marker-to-limb association, a new sample is created by either moving a marker to another limb or by switching the limbs assigned to two groups of markers (operations which are performed by either moving a non-zero element of Ω to another column or by interchanging two of its columns).

It was found that MCMC techniques did not provide good estimates of the marker-to-limb association variable, Ω. One possible reason for this was that the cost function is multi-modal and in order to sufficiently search the space of possible associations, these techniques need to run for an impractical length of time. Whereas the proposal function provides a method of moving randomly about the search space, the methods proposed in Section 3 use information such as how the relative distance between markers changed over time in order to move directly to likely associations.

Acknowledgements

We would like to thank Mark Thompson from Lund University in Sweden for supplying us with the motion capture and CT data used in testing our estimation algorithms. We would also like to thank Manuel Davey

for insightful discussion on MCMC techniques and Sahan Gamage on the least-squares procedure for estimating the model parameters.

References

[1] D.P. Bertsekas, The auction algorithm for assignment and other network flow problems: a tutorial, Interfaces 20 (1990) 133–149.
[2] S. Blackman, R. Popoli, Design and analysis of modern tracking systems, Artech House, Norwood, MA, 1999.
[3] R.B. Davis, S. Ounpuu, D. Tyburski, A gait analysis data collection and reduction technique, Hum. Mov. Sci. 10 (1991) 575–587.
[4] O.D. Faugeras, Three-dimensional Computer Vision, MIT Press, Cambridge, MA, 1993.
[5] S. Gamage, M. Ringer, J. Lasenby, Estimation of centres and axes of rotation of articulated bodies in general motion for global skeleton fitting, Technical Report CUED/_F-INFENG/_TR.408, Signal Processing Group, Engineering Department, Cambridge University, 2001.
[6] W. Gander, G.H. Golub, R. Strebel, Fitting of circles and ellipses: least squares solution, Technical report, Department Informatik, ETH, Zurich, Switzerland, 1994.
[7] D.M. Gavrila, L.S. Davis, 3-D model-based tracking of humans in action: a multi-view approach, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, USA, 1996.
[8] W.K. Hastings, Monte Carlo sampling methods using markov chains and their applications, Biometrika 57 (1970) 97–109.
[9] L. Herda, P. Fua, R. Plänkers, R. Boulic, D. Thalmann, Skeleton-based motion capture for robust reconstruction of human motion, in: Proceedings of Computer Animation, IEEE, CS Press, 2000.
[10] J. Lasenby, W.J. Fitzgerald, A.N. Lasenby, C.J.L. Doran, New geometric methods for computer vision—an application to structure and motion determination, Int. J. Comput. Vis. 26 (3) (1998) 191–213.
[11] A. Menache, Understanding Motion Capture for Computer Animation and Video Games, Morgan Kaufmann, Los Altos, CA, 2000.
[12] J. Munkres, Algorithm for the assignment and transportation problems, SIAM 5 (1957) 32–38.
[13] J.F. O'Brien, B.E. Bodenheimer, G.J. Brostow, J.K. Hodgins, Automatic joint parameter estimation from magnetic motion capture data, in: Proceedings of the Graphics Interface, Montreal, Canada, May 2000, pp. 53–60.
[14] A.B. Poore, N. Rijavec, Partitioning multiple data sets: multi-dimensional assignments and Lagrangian relaxation, in: P.M. Pardalos, H. Wolkowicz (Eds.), Quadratic assignment and related problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 16, AMS, 1994, pp. 317–342.
[15] M. Ringer, J. Lasenby, Modelling and tracking articulated motion from multiple camera views, in: Proceedings of the 11th British Machine Vision Conference (BMVC), Bristol, UK, September 2000, pp. 172–181.
[16] M. Ringer, J. Lasenby, Multiple hypothesis tracking for automatic optical motion capture, in: Proceedings of European Conference Computer Vision (ECCV), Copenhagen, Denmark, 2002.
[17] M.C. Silaghi, R. Plänkers, R. Boulic, P. Fua, D. Thalmann, Local and global skeleton fitting techniques for optical motion capture, in: N. Magnenat-Thalmann, D. Thalmann (Eds.), Modelling and Motion Capture Techniques for Virtual Environments, Number 1537 in Lecture Notes for Artificial Intelligence, 1537, Springer, Berlin, 1998, pp. 26–40.
[18] A.J. Stoddart, P. Marázek, D. Ewins, D. Hynd, A computational method for hip joint centre location from optical markers, in: Proceedings of the 10th British Machine Vision Conference (BMVC99), Nottingham, UK, 1999.
[19] L. Tierney, Markov chains for exploring posterior distributions, Ann. Stat. 22 (1994) 1701–1728.