A Procedure for Automatically Estimating Model Parameters in Optical Motion Capture

Maurice Ringer and Joan Lasenby
Engineering Dept, Cambridge University
Cambridge, CB2 1PZ, UK
{mar39,jl}@eng.cam.ac.uk

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 which limb before calculating the position of the underlying skeleton. The only assumption is that the skeleton is made up 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.

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 [5, 6, 12]. 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. Typically, these model parameters are assumed known and entered into the tracking system manually [8].

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.

BMVC 2002 doi:10.5244/C.16.73

Recently, techniques have been proposed to calculate much of this information, particularly the location of the joints [4, 10, 13, 14], however all of these require the three-dimensional trajectory of each marker and knowledge of which limb each marker is attached to. And in order to calculate the three-dimensional 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 parameters, which in turn are used to track more complex motion. Each limb should also be rotated about its joints during the sequence, so that the joint locations can be identified.

The suggestion of a training sequence to calculate model parameters can also be found in [6], however their methods are not automatic and use a non-optimal iterative technique for calculating the joint locations.

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

1. Calculate the three-dimensional 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, 3 and 4 describe these three steps in more detail.

2 Generating 3D marker positions

The first stage in estimating the model parameters is calculating the three–dimensional trajectory of each marker during the training sequence. Let the position of each marker n at time k be $y_n(k)$, where $n \in \{1, .., N\}$ and $k \in \{1, .., K\}$. 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, and estimating this association for each time frame using a global optimisation or batch technique is computationally infeasible. Instead, we propose standard sequential estimation (tracking) techniques [2] to perform this task.

Upon the arrival of a new set of detections at time k, we

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–detected-point association at time k using $\hat{y}_n(k)$.

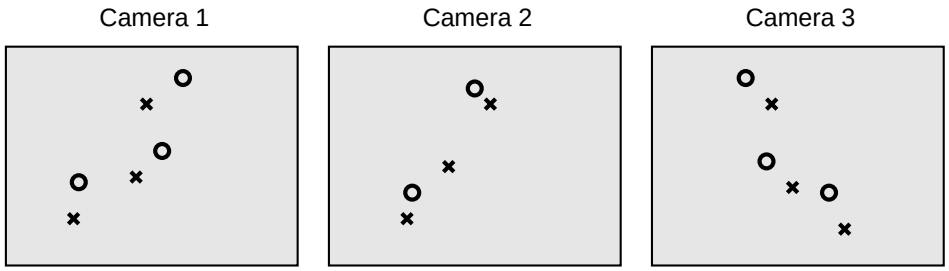

Figure 1: Example input to the 3D point tracker. The predicted positions of the markers on the camera image planes are shown by the crosses, while the circles mark the detections. The association step requires matching circles to crosses in each camera, and matching circles between cameras. Note that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

3. Calculate the three–dimensional 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 [7]. 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, $y_n(k) = \hat{y}_n(k)$.

Of these three steps, calculating the association proves the most problematic. At each time k, we desire to label each detection in each camera with the marker that generated it. The problem involves minimising the cost of associating markers to detections, but also the cost of associating a detection in one camera to a detection in another. Figure 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. $\chi_{i_0,i_1,..,i_M}$ 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, $\chi_{i_0,i_1,..,i_M}$ is 0. We require χ which minimises

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

where Q_m is the number of detections in camera m and $c_{i_0,i_1,..,i_M}$ is the cost of assuming that marker i_0 generated detections $i_1, .., i_M$ in cameras $1, .., M$ respectively.

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

where $r(i_1, .., i_M)$ is a measure of how well detections $i_1, .., i_M$ reconstruct a single point in space (that is, how well they satisfy the epipolar constraint [3]), 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 minimisation of equation (1) is done subject to the constraint that χ marginalised over any one of its dimensions is unity. This ensures that each marker is associated with at

most one detection in each camera and that any detection in a given camera is associated with at most one detection in another camera.

The problem described here is the general $(M+1)$–dimensional linear association problem. For the 2D case (for example, associating markers to the detections in a single camera, or the detections between two cameras only), fast and efficient algorithms exist for computing the optimum association [1, 9]. However, for higher dimensions ($M \geq 2$) the problem is NP-hard.

The general data association problem has recently received much attention from the radar tracking community who recommend a technique called Lagrangian relaxation for estimating the optimal association [2, 11]. 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. From the dual solution, a feasible "primal" solution can be attained by considering the two dimensions 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 it and the dual solution provides a measure of how close it is to the optimum. The solution is refined by weighting the elements of $c_{i_0,i_1,..,i_M}$ that violate the uniqueness constraint in the dual solution by some amount (the Lagrangian multipliers) and calculating new dual and primal solutions.

Using Lagrangian relaxation to estimate the most likely marker–to–detected-point 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 three–dimensional 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, $\chi_{i_1,..,i_M}$, 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 three-dimensional 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 zero otherwise. We assume that a given marker can lie only on a single limb, yet a given limb may contain any number of markers.

To ensure the system is more robust, it is desired to calculate not only a single association, but the most likely a associations, where a is typically of the order of 10 or 20. The

a proposed marker–to–limb associations are then passed to 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 true association is the one which provides the best estimate of the model parameters.

Calculating the most likely values of Ω is performed in two stages. 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. Markers are grouped as being attached to the same limb if the element of D corresponding to the variance of the distance between them is less than a certain threshold. In the event of an inconsistency, for example, if a marker is grouped with two others that are not themselves grouped together, the smaller element of D has priority. The threshold is chosen so that exactly L groups of markers are created.

This procedure creates a single set of groups of markers, or marker–to–group association. Different such associations can be created by grouping together markers that were forced to be in different groups previously to avoid inconsistencies. In these cases, the inconsistency is solved by forcing a different pair of markers to be in different groups. Let a_1 be the 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 first calculating the L_3 distance between the centroid of each group of markers in a given time frame, usually the first, for this is typically when the actor is standing in a neutral pose. For a given group–to–limb association, it is then possible to calculate the sum of the distances between the limbs that are linked. For example, the lower arm is connected to the upper arm so the distance from the centroid of the group of markers assigned as the lower arm to the centroid of markers assigned as the upper arm are included in this calculation. We select the a_2 group–to–limb associations for which the sum of the distances between the limbs is smallest, where a_2 is typically 4 or 5.

The search space for the best a_2 group–to–limb associations is lessened by considering the position in 3D space of the centroid of each group. For example, if it is known that the subject stands 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.

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. Given that we form a_2 group–to–limb associations for each, we provide, as expected, $a = a_1 a_2$ possible values of Ω for input to the algorithms of the next section.

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, $P(\Omega|y)$. 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. It is for this reason that we calculate multiple associations.

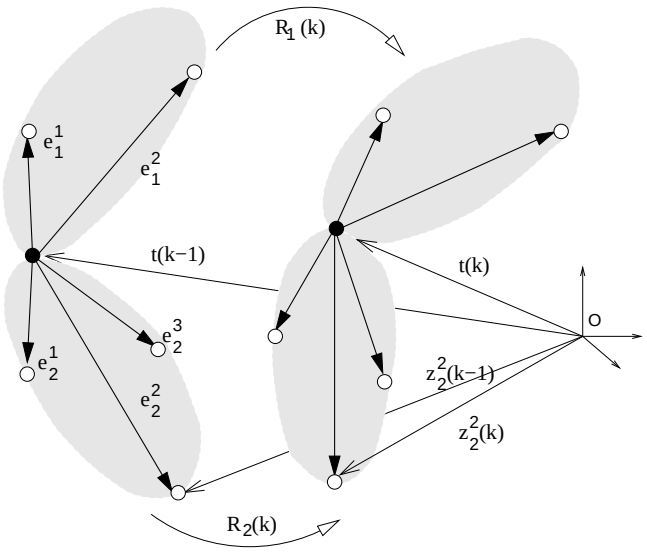

Figure 2: The motion of two limbs connected by a single ball joint. e is the marker positions relative to the joint, z is the observed position of the markers, R is the rotation of the limbs and t is the position of the joint.

The following section describes how to estimate R, e and t so that the first of the above terms, $P(R, e, t|\Omega, y)$, is maximised. This is done for each marker–to–limb association, providing a points on the surface of the posterior, one of which is its maximum.

4 Calculating the Marker Offset Vectors

Given the three-dimensional 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 three-dimensional positions of the joints around which the rotations occur and the position of the markers with respect to these joints [4, 10, 13, 14]. We adopt a similar approach to that of [4], as the solutions are closed-form and thus quick to calculate.

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 three-dimensional 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. Figure 2 illustrates this motion of the joint and two limbs.

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

combining equations of different l, and finally t is determined by substituting the estimates for R and e back into equation (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 [7] is used.

The reader is referred to [4] 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,

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

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 (primarily arms) slowly and ensuring that rotation occurred at each joint.

In each case, the Lagrangian relaxation technique of section 2 always converged on the correct marker–to–detected-point association and the 3D point tracker successfully generated trajectories of the markers, $y_n(k)$. Also for each case, the algorithms of section 3 determined the correct marker–to–limb association.

Figure 3 shows the result of calculating the marker positions relative to the joint, e, and the joint position, t, for the two limb case. Figure 4 shows the result of applying the proposed technique to a more complex skeletal model. In this case, two arms and 15 markers were being tracked and used to calculate the required model parameters (the axis between the shoulders was considered as a limb).

As can be seen from these two figures, the resulting estimates of the marker, joint and limb positions fit the observed marker positions well, and the estimated underlying skeleton appears to follow the motion of the subject.

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

6 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 runs very fast.

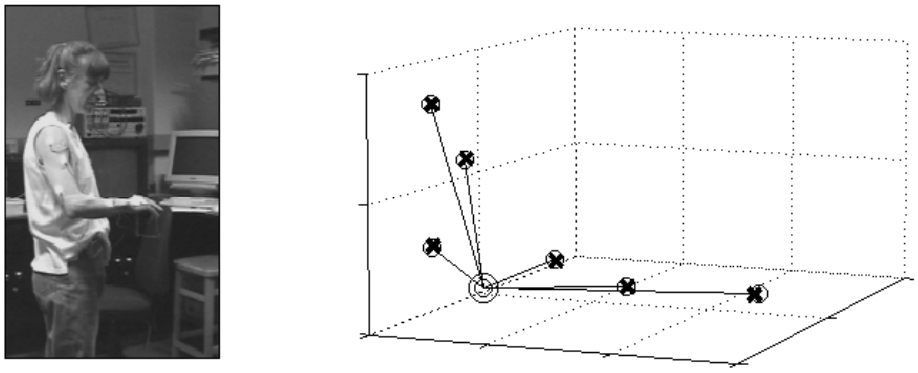

Figure 3: Calculating the marker offset positions, e (the circles and lines) and centre of rotation, $t(k)$ (double circle) for two limbs. Crosses represent the observed marker positions, $z(k)$.

References

[1] D. P. Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20:133–149, 1990.

[2] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, 1999.

[3] O. Faugeras. *Three-dimensional Computer Vision*. MIT Press, 1993.

[4] S. Gamage, M. Ringer, and 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, Dept of Engineering, Cambridge Univ, Cambridge, UK, 2001.

[5] D. M. Gavrila and L. S. Davis. 3-D model–based tracking of humans in action: A multi-view approach. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR96)*, San Francisco USA, 1996.

[6] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann. Skelton–based motion capture for robust reconstruction of human motion. In *Proc. Computer Animation*, IEEE CS Press, 2000.

[7] J. Lasenby, W.J. Fitzgerald, A.N. Lasenby, and C.J.L. Doran. New geometric methods for computer vision – an application to structure and motion determination. *International Journal of Computer Vision*, 26(3):191–213, 1998.

[8] A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Korgan Kaufmann Publishers, Academic Press, 2000.

[9] J. Munkres. Algorithm for the assignment and transportation problems. *SIAM*, 5:32–38, 1957.

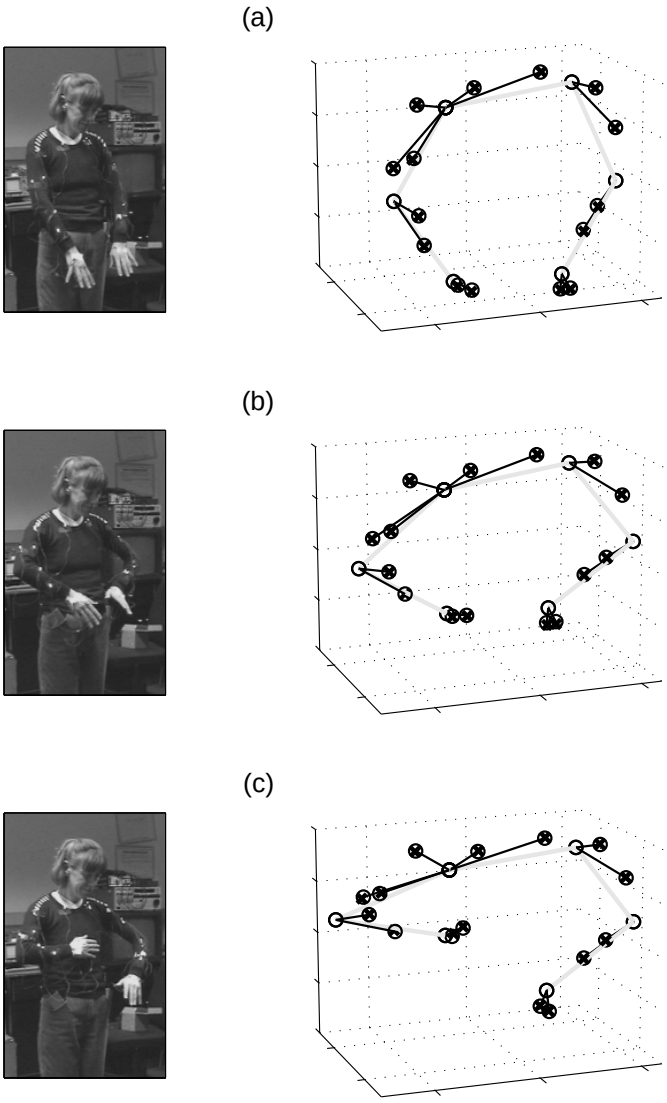

Figure 4: The calculated underlying skeleton for a seven limb model, shown at three different time frames during the training sequence. The gray lines represent the estimated limbs, the circles and lines represent the marker offset vectors, and the crosses represent the observed marker positions, $z(k)$.

[10] J. F. O'Brien, B. E. Bodenheimer, G. J. Brostow, and J. K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proc. Graphics Interface*, pages 53–60, Montreal, Canada, May 2000.

[11] A. B. Poore and N. Rijavec. Partitioning multiple data sets: Multidimensional assignments and Lagrangian relaxation. In *Quadratic Assignment and Related Problems, P. M. Pardalos and H. Wolkowicz, eds.*, pages 317–342. DIMACS Series in Discrete Mathematics and Theoretical Computer Science 16, AMS, 1994.

[12] M. Ringer and J. Lasenby. Modelling and tracking articulated motion from multiple camera views. In *Proc 11th British Machine Vision Conference (BMVC2000)*, volume 1, pages 172–181, Bristol, UK, September 2000.

[13] M.C. Silaghi, R. Plänkers, R. Boulic, P. Fua, and D. Thalmann. Local and global skeleton fitting techniques for optical motion capture. In *Modelling and Motion Capture Techniques for Virtual Environments, N. Magnenat-Thalmann and D. Thalmann, Eds.*, number 1537 in Lecture Notes for Artificial Intelligence, pages 26–40. Springer, 1998.

[14] A. J. Stoddart, P. Marázek, D. Ewins, and D. Hynd. A computational method for hip joint centre location from optical markers. In *Proc 10th British Machine Vision Conference (BMVC99)*, Nottingham, UK, September 1999.