

Real-time marker prediction and CoR estimation in optical motion capture

Andreas Aristidou · Joan Lasenby

Published online: 31 January 2012
© Springer-Verlag 2012

Abstract Optical motion capture systems suffer from marker occlusions resulting in loss of useful information. This paper addresses the problem of real-time joint localisation of legged skeletons in the presence of such missing data. The data is assumed to be labelled 3d marker positions from a motion capture system. An integrated framework is presented which predicts the occluded marker positions using a Variable Turn Model within an Unscented Kalman filter. Inferred information from neighbouring markers is used as observation states; these constraints are efficient, simple, and real-time implementable. This work also takes advantage of the common case that missing markers are still visible to a single camera, by combining predictions with under-determined positions, resulting in more accurate predictions. An Inverse Kinematics technique is then applied ensuring that the bone lengths remain constant over time; the system can thereby maintain a continuous data-flow. The marker and Centre of Rotation (CoR) positions can be calculated with high accuracy even in cases where markers are occluded for a long period of time. Our methodology is tested against some of the most popular methods for marker prediction and the results confirm that our approach outperforms these methods in estimating both marker and CoR positions.

Electronic supplementary material The online version of this article (doi:10.1007/s00371-011-0671-y) contains supplementary material, which is available to authorized users.

A. Aristidou (✉) · J. Lasenby
Department of Engineering, University of Cambridge,
Cambridge, CB2 1PZ, UK
e-mail: aa462@cam.ac.uk

J. Lasenby
e-mail: jl221@cam.ac.uk

Keywords Computer vision · Filtering · Marker prediction · Joint localisation · Motion capture · Inverse kinematics

1 Introduction

Optical motion capture is a technology used to turn the observations of a moving subject (taken from a number of cameras) into 3d position and orientation information about that subject. It is commonly used to better analyse techniques for sports training and performance [28]; for observation of asymmetries and abnormalities in rehabilitation medicine [9]; in biomechanics labs (prosthetics, ergonomics); and for visualisation of virtual characters for films and computer games [48]. In general, to achieve accurate skeletal reconstruction of any legged body, 3 markers must be available on each limb segment at all times. However, even with many cameras, there are instances where occlusion of markers by elements of the scene leads to missing data. In order to unambiguously establish its position, each marker must be visible to at least two cameras in each frame. Although many methods have been developed to handle the missing marker problem, most of them are not applicable in real-time, are usually limited to short time period occlusions, and often require manual intervention. Real-time gap filling is very important in motion capture technology since the skeleton of the character can be produced simultaneously with tracking the motion; it is essential, especially in applications where interaction between the user and the computer is needed.

This paper investigates methodologies for real-time marker prediction, under multiple cases of occlusion, to drive centre of rotation (CoR) estimates, and then to automatically establish the skeleton model. A real-time integrated framework is presented, which predicts the occluded

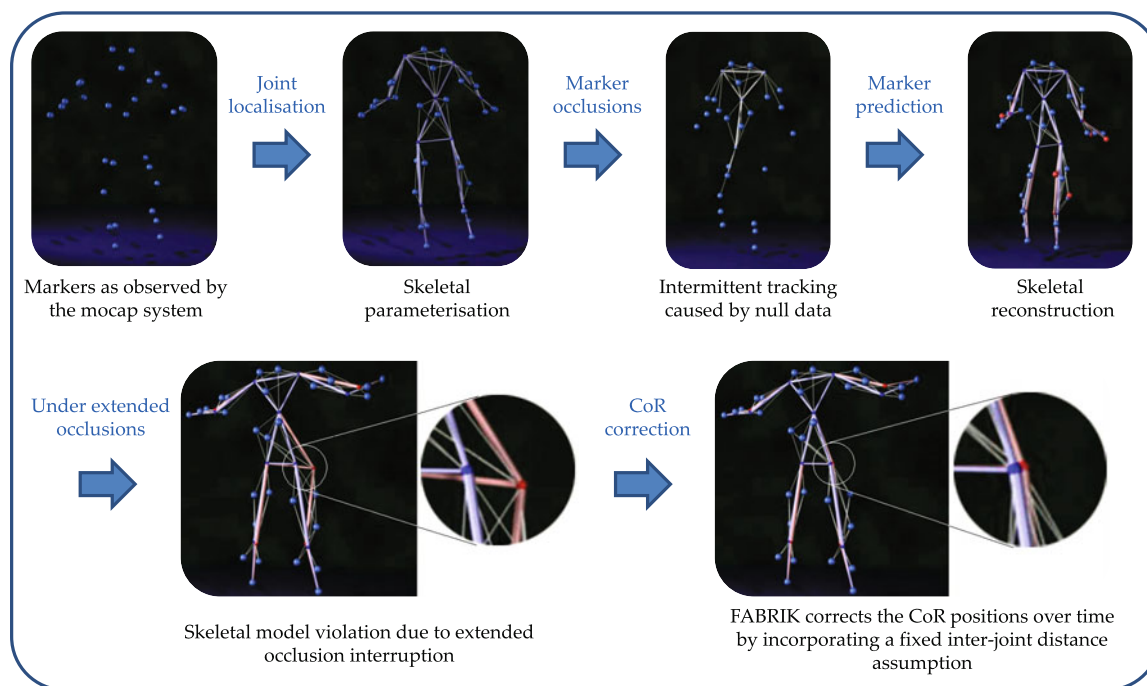


Fig. 1 An overview of the proposed method used to compensate for interruptions caused by occlusions and to drive real-time skeletal parameterisations

marker positions using a Variable Turn Model within an Unscented Kalman filter (UKF). The previous marker positions are used within the framework in addition to information related to the missing markers of the current frame, inferred from an approximate rigid body assumption. The predicted marker positions are then used to locate the joints. Without assuming any skeleton model, we take advantage of the fact that for markers on a given limb segment, the inter-marker distance is approximately constant. The proposed marker constraint methodology is simple, real-time implementable, and very efficient. Our method is automatic and scalable, without requiring any parameters to be set by the user. It considers all the cases of marker occlusion within a limb resulting in accurate predictions even in cases where all markers on a limb segment are missing for an extended period of time. The proposed approach is the first method that takes advantage of the special, but common, case where missing markers are visible to just one camera, reducing the marker estimation error significantly. With a continuous stream of accurate labelled 3d data, we can perform real-time CoR estimation; the CoR position is thereafter corrected via a real-time Inverse Kinematic technique which guarantees that the inter-joint pairwise distances remain constant over time. To the best of our knowledge, this is the first methodology that operates inter-joint constraints, using real-time Inverse Kinematic techniques, in order to maintain fixed bone lengths over time in optical motion capture systems. A skeletal reconstruction is thereby achieved,

producing information which can be used for visual performance feedback. Figure 1 shows the outline organisation of our system. Experiments demonstrate that our methodology effectively recovers good estimates about the true positions of the missing markers and CoRs, agreeing with human intuition, even if all the markers on a limb are occluded for a long period of time. Our work has been tested against some of the most popular methods for compensating for interruptions caused by marker occlusion and the results confirm that our method generally outperforms others. The movements produced are smooth and are without abnormalities or oscillations, resulting in natural reconstructed motion.

2 Related work

Many papers have focused on methods for joint localisation. *Sphere fitting* approaches are the most commonly used methods for calculating the CoR. This group of methods assumes that all markers remain a constant distance from the centre of rotation. Silaghi et al. in [59] use the Levenberg–Marquardt method to optimise the CoR location and the radii of the marker spheres, via a cost function which sums a per marker cost over all markers and all frames. Halvorsen et al., in [27], describe a closed form solution using the geometric properties of the sphere. In [24], Gamage and Lasenby also introduce a closed form solution, using a cost function of the squared differences in the squared distance

from the CoR to a marker and the radius of the sphere associated with that marker. An alternative approach provided by Halvorsen, [26], gives a Bayesian analysis of the algorithm of [24], providing a first-order approximation of the effect of isotropic Gaussian noise upon the algorithm.

Another group of methods is that termed *Transformational techniques*; they assume that markers are rigidly attached to limb segments. Such an approach was implemented in [33, 51], and [22], where the limb orientation was obtained from sets of optical markers. In [12], a sequential algorithm was presented to locate the rotation centres of a human skeleton from marker data assuming that all markers are attached to a rigid body. The method is closed form, thus enabling real-time implementation.

Whilst several methods to estimate the location of the CoR have been proposed, the performance of most is unsatisfactory in the presence of unusual motions or of many contiguous occlusion-affected frames. Indeed, there are instances, even with expensive motion capture systems, where occlusion of markers by elements of the scene leads to missing data. While the discarding of frames with missing markers is a common technique, the omission of specific data could lead to the loss of useful information. Long-running occlusions leading to a large sequence of missing data can also cause complete failure of the system. Several methods have been proposed to predict the occluded markers in order to drive CoR estimation and skeletal reconstruction. Interpolation of the data using linear or nonlinear approaches is commonly used [58, 69]; this can produce accurate results, but it is a post-processing technique requiring data prior to and after the occlusion. Recently, Piazza et al. in [55] presented an extrapolation algorithm which assumes that the most common motion behaviors are circular or linear movements; however, this method can produce reliable predictions only for a limited number of occluded frames.

Rhijn and Mulder, [56], proposed a model-based optical tracking and model estimation system for composite interaction devices; however, it is an off-line procedure unsuitable for real-time applications. Dorfmueller in [20] used an extended Kalman filter (EKF) to predict the missing markers using previously available marker information, while Welch et al. in [68] used an EKF to resolve occlusions based on the skeletal model of the tracked person. Maida et al. in [47] prevent intermittent tracking by taking measurements from an optical flow method to supply a hybrid approach based on Kalman filtering. For this reason, an additional marker is introduced for the optical flow tracker; nevertheless, it is impossible to track the object and maintain virtual graphics overlaying when both markers are not identified. Tak and Ko, [63], employed an Unscented Kalman Filter to ensure that the motion capture data remains in a feasible set. However, these methods require manual intervention or become ineffective in cases where markers are

missing for an extended period of time. Aristidou et al. [1] also presents an EKF method using a constant velocity (CV) model with marker constraints from neighbouring¹ markers. However, the CV model limits its use to problems with constant marker velocity. These methods also do not take into consideration the fact that bones are rigid, thus the inter-joint pairwise distances should remain constant over time. Li et al. [41] propose DynaMMo, an approach that uses a Linear Dynamical System (LDS) to model motion capture data under sequences with missing values; in [42], the same authors introduce BoLeRo, a similar technique which also takes into consideration bone length constraints. The suggested algorithm has a ‘hard’ and ‘soft’ version of bone constraints assuming rigidity limitations of the distance between markers on a given segment limb. Although their algorithm results in smooth motion, the method is complex and expensive in terms of computational cost.

Herda et al., in [30] and [31], used a post-processing approach to increase the robustness of a motion capture system by using a sophisticated human model. The neighbouring markers that share kinematic relations with the occluded markers were used to help the estimation of the missing markers. However, the skeleton information must be known a priori in order to apply this method. Hornung and Sar-Dessai, [35], also developed a system to compensate for interruptions caused by occlusions by performing constant inter-distance signatures between neighbouring markers. Nevertheless, this approach may become ineffective when all or a significant number of markers are missing so that no information on that limb can be inferred from the available LEDs. Ringer and Lasenby, [57], also present an automatic method to identify indistinguishable markers based on cliques.² However, this requires an off-line procedure in order to determine marker cliques and parameters of the skeletal structure. Zordan and Van Der Horst in [72] mapped 3D marker position data to joint trajectories for a fixed limb-length skeleton, by attaching virtual springs and controllers, to follow their Cartesian positions. In general, these skeleton methods could work well for short time occlusions but fail to track the missing markers for large occluded sequences.

In [25], a style-based inverse kinematic method has been developed where a Gaussian Process Latent Variable Model (GPLVM) was used along with a pre-specified kinematic model. Wang et al., [66], presented a Gaussian process dynamical model (GPDM) in order to learn the human pose and motion models. They observed the motion using a chain of latent variables and nonlinear mapping from the latent space; the proposed learned model was also able to cope

¹Neighbours are the markers belonging to the same limb segment.

²Markers in a clique have constant distances between each other.

with marker occlusions. Although these are real-time processing methods, they require knowledge of skeleton information which severely restricts their use. Chai and Hodgins, [13], present a method that uses the neighbouring markers to estimate the missing marker in the current frame. They propose a local linear model from these neighbours and then reconstruct the full pose of the frame by conducting an optimisation in the space constrained by a pre-recorded database. Yu et al., [71], proposed an online motion capture labelling approach which also recovers missing markers. They cluster the markers into a number of rigid bodies based on the standard deviations of the marker-pair distances and if their fitting-rigid-bodies algorithm did not classify all the markers into rigid bodies, a missing marker auto-recovery method is applied assuming that the inter-marker distances are fixed over time. However, a training session is needed, the auto-recovery method for marker estimation does not take into account the limb segment rotation, no information about markers visible to a single camera is considered and the CoR estimation is not investigated under marker occlusions.

Park and Hodgins, [52], fill the missing data by learning a statistical model of the spatial relationship between each marker and its neighbours; they use a Principal Component Analysis (PCA) on each marker position and its neighbours throughout the entire motion. In [64], the authors modelled the human motion and filled the gaps in the data using a Conditional Restricted Boltzmann Machine (CRMB) with discrete hidden states. Their approach was trained using non-linear binary representations, conditioned on previous frames; at the same time, they took into consideration the correlation between joint angles, to produce more accurate results. Liu et al., [45, 46], presented a piecewise linear approach for estimating human motions from a pre-selected set of informative markers (principal markers). A pre-trained classifier identifies an appropriate local linear model for each frame. Missing markers are then recovered using available marker positions and the principal components of the associated model. In [36], the data were mapped onto a target motion by searching over patterns in existing databases. Recently, Courty and Cuzol in [16] presented an optimal data interpolation method through conditional stochastic simulation for automatic motion completion. However, this data-driven family of methods requires an off-line training procedure and the results are highly dependent on training data and limited to those models and movements the system has been trained on.

In this work, we prevent intermittent tracking using a UKF model which uses the velocity and acceleration derived from marker positions as the observation sequence, in addition to a constant rotation model. Inferred information from neighbouring markers has been utilised for a better approximation of the marker estimates. Having a continuous

flow of data, we can drive real-time skeletal parameterisation. The CoR positions are thereafter corrected using an adapted version of the FABRIK Inverse Kinematics solver, which uses an inter-joint constant distance signature. So far, no existing method operates such bone length constraints to prevent rigid body violation. There are, however, methods where the chosen skeletal parameterisation is composed of rotations, thus no inter-joint kinematic restrictions are required [18, 30].

3 Calculating the centre of rotation

During capture, markers must be carefully placed on the body in order to obtain good results. Results using markers placed too close to the CoR are more susceptible to errors since a small error may cause large deviations in the estimated rotation, leading to erroneous calculation of the model parameters. The data discussed here are labelled from an active marker system (PhaseSpace [54]) where no tracking is necessary. The PhaseSpace impulse system identifies and tracks individual markers from their unique modulation and in this paper problems related to marker inversion are not therefore considered. In general, 3 markers per bone segment are required to estimate the CoR for joints with 3 Degrees of Freedom (DoF); for simpler problems having fewer DoF, such as knees and elbows, the CoR can be calculated with fewer markers [14]. In this paper, we consider the general case of joints with 3 DoF since no prior knowledge of the model or joint-type is assumed.

Locating the CoRs is a crucial step in acquiring a skeleton from raw motion capture data. To calculate the joints between two sets of markers, it is helpful to have the rotation of a limb at any given time. We can estimate the orientation of a limb at time k relative to a reference frame using the *Procrustes* formulation [34].

The location of the joints can be calculated using the approach in [12]. This takes advantage of the approximation that all markers on a segment are attached to a rigid body. Suppose the markers are placed on two segments (x and y) joined by a CoR. Let the CoR location in frame k be \mathbf{C}_k . The vectors from the CoR to markers in the reference frame are denoted by \mathbf{a}_x^i and \mathbf{a}_y^j for limbs x and y , respectively, where i and j are marker labels. The positions of the markers in frame k are given by:

$$\mathbf{x}_i^k = \mathbf{C}^k + R_x^k \mathbf{a}_x^i \tilde{R}_x^k \quad \mathbf{y}_j^k = \mathbf{C}^k + R_y^k \mathbf{a}_y^j \tilde{R}_y^k \quad (1)$$

where R_x and R_y are the rotors (quaternions) expressing the rotation of the joint limbs x and y , respectively. \tilde{R} is the quaternion conjugate of R . Let \mathbf{b}_{ij}^k be the vector from \mathbf{x}_i^k to \mathbf{y}_j^k , that is

$$\mathbf{b}_{ij}^k = \mathbf{x}_i^k - \mathbf{y}_j^k = R_x^k \mathbf{a}_x^i \tilde{R}_x^k - R_y^k \mathbf{a}_y^j \tilde{R}_y^k \quad (2)$$

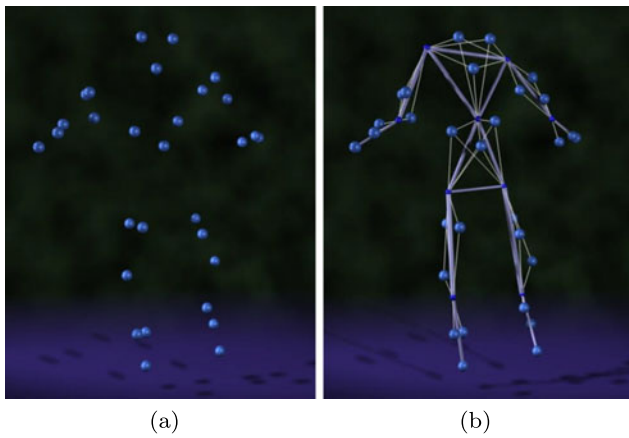


Fig. 2 An example of CoR estimation of a human body model using [12]. (a) The marker positions as returned by the motion capture system, (b) the calculated CoRs and the skeletal reconstruction

A cost function S can be constructed that has a global minimum at the correct values of \mathbf{a}_x^i and \mathbf{a}_y^j if the data is noise free, and returns a good estimate in the presence of moderate noise.

$$S = \sum_{k=1}^m \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} [\mathbf{b}_{ij}^k - (R_x^k \mathbf{a}_x^i \tilde{R}_x^k - R_y^k \mathbf{a}_y^j \tilde{R}_y^k)]^2 \quad (3)$$

where n_x, n_y are the number of markers on limbs x and y , respectively, and m is the number of frames used for the calculations. The minimum is given by the solution of the simultaneous linear equations, obtainable by differentiation:

$$\mathbf{a}_x^i = \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k \tilde{\mathbf{b}}^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \tilde{\mathbf{a}}_y \tilde{R}_y^k R_x^k \quad (4)$$

$$\mathbf{a}_y^j = \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k \tilde{\mathbf{b}}^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \tilde{\mathbf{a}}_x \tilde{R}_x^k R_y^k \quad (5)$$

where

$$\tilde{\mathbf{b}}^k = \frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \mathbf{b}_{ij}^k \quad \tilde{\mathbf{a}}_w = \frac{1}{n_w} \sum_{i=1}^{n_w} \mathbf{a}_w^i \quad w = \{x, y\}$$

Having calculated the R_w^k and $\tilde{\mathbf{a}}_w$, we can locate the CoR. Figure 2 demonstrates an example of CoR estimation and skeletal reconstruction in real-time using the above method. However, due to occlusions, there are instances where not all marker positions are available. If all markers are available on one limb segment, w , the CoR may be estimated using only the current R_w^k and $\tilde{\mathbf{a}}_w$ as estimated in the previous frame, when all markers were visible, via (1). If there are markers occluded on both limb segments, a marker prediction methodology is needed.

4 Marker prediction

The estimates for marker positions can be predicted using filtering. In this process, each single marker can be tracked individually and constraints for neighbouring markers can be incorporated. Most tracking problems require a dynamic model for accurate estimation of the trajectory of a maneuverable object. During the last decades, various mathematical models of target motions have been developed. Singer [60, 61] proposes a model which assumes that the target acceleration is a zero-mean first-order stationary Markov process. Based on *Singer's* assumption, many papers have proposed a constant or variable acceleration model (e.g. [43]).

The *nearly constant turn* NCT and *constant velocity* CV models are based on a constant-speed condition and constant turn-rate assumption which restricts the variety of possible supported maneuvers. Within this work, a Variable Turn Model (VTM) [5] is implemented which uses the velocity and acceleration (it is assumed that the target's velocity and acceleration are not constant over time) of the tracked object as state parameters of the Unscented Kalman filter in addition to rotation updates assuming that the rotation between two consecutive frames remains constant.

Unscented Kalman filter Kalman filtering has been extensively used for real time estimation of linear dynamic systems. However, the traditional Kalman filter [40] is not suitable for use with non-linear dynamical systems, even if Gaussian approximations to the joint distribution of state \mathbf{x} and measurement \mathbf{y} are made. The Extended Kalman Filter (EKF) [38] is a minimum mean-square-error (MMSE) estimator which extends the scope of the Kalman filter to non-linear optimal filtering problems. It forms a Gaussian approximation to the joint distribution of state and measurement using a Taylor series-based transformation. Nevertheless, EKF implementation is complex (Jacobian and Hessian matrices with second-order filters are required), difficult to tune, and only reliable for systems that are almost linear on the timescale of the updates.

The Unscented Kalman Filter (UKF), [39], propagates mean and covariance information through nonlinear transformations providing more accurate results than the EKF, for a similar computational cost. Consider propagating an n_x -dimensional random variable \mathbf{x} and assume \mathbf{x} has mean $\bar{\mathbf{x}}$ and covariance \mathbf{P}_x . First, a set of $2n_x + 1$ weighted samples or *sigma points* $\mathcal{S}_i = \{W_i, \mathcal{X}_i\}$ are deterministically chosen so that the true mean and covariance of the random variable \mathbf{x} can be completely recovered from them. A set of scaled sigma points $\mathcal{S} = \{\mathbf{W}, \mathcal{X}\}$ can be calculated by setting:

$$\lambda = \alpha^2(n_x + \kappa) - n_x \quad (6)$$

and selecting the sigma point set by

$$\begin{aligned} W_0^{(m)} &= \lambda / (n_x + \lambda) & i &= 0 \\ W_0^{(c)} &= \lambda / (n_x + \lambda) + (1 - \alpha^2 + \beta) & i &= 0 \\ W_i^{(m)} &= W_i^{(c)} = 1 / \{2(n_x + \lambda)\} & i &= 1, \dots, 2n_x \end{aligned}$$

where α is a positive scaling parameter which controls the size of the sigma point distribution. $\kappa \geq 0$ is also a scaling parameter and W_i is the weight associated with the i^{th} point such that $\sum_{i=0}^{2n_x} W_i = 1$. Note that the (m) and (c) superscripts are just used for indexing the weight W_i . Merwe et al., [49], proposed $\kappa = 0$, to guarantee positive semidefiniteness of the covariance matrix and $0 \leq \alpha \leq 1$ to avoid sampling non-local effects when the nonlinearities are strong. $\beta \geq 0$ is a weighting term which incorporates knowledge of the higher order moments of the distribution. $\beta = 2$ is the optimal choice for a Gaussian prior.

Let the original state and noise variables at time k be $\mathbf{x}_k^\alpha = [\mathbf{x}_k^T \mathbf{v}_k^T \mathbf{n}_k^T]^T$. The sigma point selection scheme is applied to this augmented state Random Variable (RV) to calculate the corresponding sigma matrix, \mathcal{X}_k^α . The mean $\bar{\mathbf{x}}$ and covariance \mathbf{P} of the Gaussian approximation is updated to the posterior distribution of the states as follows:

$$\begin{aligned} \bar{\mathbf{x}}_0 &= E[\mathbf{x}_0] \\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T] \\ \bar{\mathbf{x}}_0^\alpha &= E[\mathbf{x}^\alpha] = [\mathbf{x}_0^T \ 0 \ 0]^T \\ \mathbf{P}_0^\alpha &= E[(\mathbf{x}_0^\alpha - \bar{\mathbf{x}}_0^\alpha)(\mathbf{x}_0^\alpha - \bar{\mathbf{x}}_0^\alpha)^T] = \begin{bmatrix} \mathbf{P}_0 & 0 & 0 \\ 1 & \mathbf{Q} & 0 \\ 1 & 0 & \mathbf{R} \end{bmatrix} \end{aligned} \quad (7)$$

For $k \in \{1, \dots, \infty\}$ the sigma points are equal to

$$\mathcal{X}_{k-1}^\alpha = [\bar{\mathbf{x}}_{k-1}^\alpha \ \bar{\mathbf{x}}_{k-1}^\alpha \pm \sqrt{(n_\alpha + \lambda)\mathbf{P}_{k-1}^\alpha}] \quad (8)$$

and the time update is given by

$$\begin{aligned} \mathcal{X}_{k|k-1}^x &= \mathbf{f}(\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^v) \\ \bar{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{2n_a} W_i^{(m)} \mathcal{X}_{i,k|k-1}^x \\ \mathbf{P}_{k|k-1} &= \sum_{i=0}^{2n_a} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^x - \bar{\mathbf{x}}_{k|k-1}] [\mathcal{X}_{i,k|k-1}^x - \bar{\mathbf{x}}_{k|k-1}]^T \\ \mathcal{Z}_{k|k-1} &= \mathbf{h}(\mathcal{X}_{k|k-1}^x, \mathcal{X}_{k-1}^n) \\ \bar{\mathbf{z}}_{k|k-1} &= \sum_{i=0}^{2n_a} W_i^{(m)} \mathcal{Z}_{i,k|k-1}^x \end{aligned}$$

where $\mathbf{f}(\cdot)$ is the transition and $\mathbf{h}(\cdot)$ the observation function. The measurement update equations are:

$$\begin{aligned} \mathbf{P}_{\bar{\mathbf{z}}_k \bar{\mathbf{z}}_k} &= \sum_{i=0}^{2n_a} W_i^{(c)} [\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_{k|k-1}] [\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_{k|k-1}]^T \\ \mathbf{P}_{x_k z_k} &= \sum_{i=0}^{2n_a} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^x - \bar{\mathbf{x}}_{k|k-1}] [\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_{k|k-1}]^T \\ \mathbf{K}_k &= \mathbf{P}_{x_k y_k} \mathbf{P}_{\bar{\mathbf{y}}_k \bar{\mathbf{y}}_k}^{-1} \\ \bar{\mathbf{x}}_k &= \bar{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{Z}_k - \bar{\mathbf{z}}_{k|k-1}) \\ \mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\bar{\mathbf{z}}_k \bar{\mathbf{z}}_k} \mathbf{K}_k^T \end{aligned}$$

where $\mathbf{x}^\alpha = [\mathbf{x}^T \mathbf{v}^T \mathbf{n}^T]^T$, $\mathcal{X}^\alpha = [(\mathcal{X}^x)^T (\mathcal{X}^v)^T (\mathcal{X}^n)^T]^T$, λ is the composite scaling parameter, $n_\alpha = n_x + n_v + n_n$, \mathbf{Q} is the process noise covariance, \mathbf{R} is the measurement noise covariance, \mathbf{K} is the Kalman gain, W_i are the Unscented Transform weights, and \mathbf{Z}_k is the observation vector.

The transition function $\mathbf{f}(\cdot)$ and the observation function $\mathbf{h}(\cdot)$ are very important for implementing efficient UKF filtering. In this model, the transition function is taken to be a Variable Turn Model (VTM) with target velocity and acceleration the equivalents of the relevant tracked marker. However, a detailed look at real marker data indicates that the estimated velocities are invariably noisy. Many factors contribute to marker position noise, such as optical measurement noise, mis-calibration of the optical systems, reflections, motion of markers relative to the skin and motion of the skin relative to the rigid body (underlying bone). As a result, the target acceleration is mostly noisy. One method of measuring accelerations would be to attach accelerometers to the markers placed on the limb segments. Such a system, which synchronises measurements from the accelerometer and active markers was described and investigated in [6]. While the system shows some promise, we are faced with dealing with both the noise on the accelerometer readings and the fact that the reference frame of those readings must be determined. Given these difficulties, we decided to use isolated estimates of accelerations obtained from the position data. The target velocity was smoothed using a real-time median filter, thus allowing an adequate calculation of the acceleration which was therefore straightforwardly applied in the proposed model.

On the other hand, for the observation function we use a simple model which assumes that the rotation between two consecutive frames is constant. Thus, the time update of the observation state is equal to

$$\mathcal{Z}_{k|k-1} = \mathbf{R}^{k-2,k-1} \mathcal{X}_{k|k-1}^x \tilde{\mathbf{R}}^{k-2,k-1} \quad (9)$$

where $\mathbf{R}^{p,q}$ is the rotor for the rotation between frames p and q , assuming that the rotation of the markers between two consecutive frames remains constant. It is important at

this point to recall that the rotation between the previous and the current frame is re-calculated every frame just after the marker prediction using the estimated marker positions.

Early numerical implementations using Particle Filtering [21] showed that, although it could improve the performance of our state estimates, this improvement was marginal compared to the UKF-VTM and, therefore, not worth the added computational effort.

5 Applying marker constraints

The observation vector, \mathbf{Z}_k^j , gives the observed position of the tracked marker j when available, otherwise it represents estimated position. The state vector represents true position and velocity as given above. To cope with cases where markers are missing for long periods of time, a tracker that uses information from both the previous frames and the current positions of neighbouring visible markers has been implemented, taking into account the rigidity of the body. Assuming that there are three markers on each limb and the inter-marker distance is constant over time, the observation vector can be updated as given below for 5 different scenarios. The proposed marker constraints from inferred neighbouring markers are simple, real-time implementable and do not assume prior knowledge other than fixed inter-marker distance (the model is rigid). This assumption is true in a noise free environment, however, since limb segments rapidly change direction, there is noise on marker motion relative to the skin and motion of the skin relative to the rigid body (underlying bone).

5.1 All markers are visible on a given limb

Where all markers are visible on a given limb, then

$$\mathbf{Z}_k^j = H\mathbf{x}_j^k + \mathbf{v}_j^k \tag{10}$$

where \mathbf{x}_j^k is the current state of a tracked marker j on the limb, H is the *observation model* (in this case the identity) and \mathbf{v}_j^k is the observation noise. \mathbf{v} is assumed to be zero mean multivariate normal with covariance R .

5.2 One missing marker on a limb segment

In the case where two markers are visible on the limb,

$$\mathbf{Z}_k^1 = H\hat{\mathbf{x}}_1^k + \mathbf{v}_1^k \tag{11}$$

where $\hat{\mathbf{x}}_1^k$ is the estimated position of the occluded marker m_1 in frame k (assuming, in this example, that m_1 is the missing marker). $\hat{\mathbf{x}}_1^k$ can be calculated as given below. Firstly, we calculate $\mathbf{D}_{1,2}^{k-1}$ and $\mathbf{D}_{1,3}^{k-1}$ which correspond to

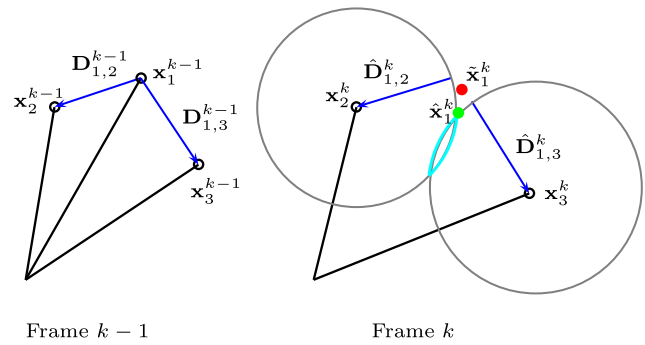


Fig. 3 The observation vector in the case of 2 visible markers. The red dot, $\tilde{\mathbf{x}}_1^k$, represents the average value as given in (14). The green dot, $\hat{\mathbf{x}}_1^k$, is the point on the intersection of the 2 spheres which is closest to $\tilde{\mathbf{x}}_1^k$

the vectors between marker m_1 and markers m_2, m_3 in frame $k - 1$, respectively. These vectors are given by

$$\mathbf{D}_{i,j}^{k-1} = \mathbf{x}_j^{k-1} - \mathbf{x}_i^{k-1} \tag{12}$$

Thereafter, these vectors are rotated as

$$\hat{\mathbf{D}}_{i,j}^k = R^{k-2,k-1} \mathbf{D}_{i,j}^{k-1} \tilde{R}^{k-2,k-1} \tag{13}$$

assuming that the rotation between the current and the previous frame is the same as that between the previous 2 frames. Predicting the current rotation using previous rotations offers marginal improvement to the system performance, and such a small improvement means that it is not worth the additional computational cost [4]. One obvious way to proceed is to calculate the point $\tilde{\mathbf{x}}_1^k$ which is an average of the estimated positions in frame k using the $\hat{\mathbf{D}}$ vectors:

$$\tilde{\mathbf{x}}_1^k = \frac{(\mathbf{x}_2^k - \hat{\mathbf{D}}_{1,2}^k) + (\mathbf{x}_3^k - \hat{\mathbf{D}}_{1,3}^k)}{2} \tag{14}$$

where \mathbf{x}_i^k is the position of marker i in frame k . We now improve on this estimate by finding the solution of the intersection of the two spheres in frame k with centres $\mathbf{x}_2^k, \mathbf{x}_3^k$ and radii $|\hat{\mathbf{D}}_{1,2}^k|$ and $|\hat{\mathbf{D}}_{1,3}^k|$, respectively. $\hat{\mathbf{x}}_1^k$ is assigned as the closest point on the circle of intersection to $\tilde{\mathbf{x}}_1^k$. Figure 3 illustrates this process.

5.3 Two missing markers on a limb segment

In the case of only one marker visible (for example m_2) on a given limb, the observation vector is given as

$$\mathbf{Z}_k^j = H\hat{\mathbf{x}}_j^k + \mathbf{v}_j^k \tag{15}$$

where $\hat{\mathbf{x}}_j^k$ is the estimated position of the occluded marker m_j ($j = 1, 3$) in frame k . $\hat{\mathbf{x}}_j^k$ is given by

$$\hat{\mathbf{x}}_j^k = \mathbf{x}_2^k - \hat{\mathbf{D}}_{j,2}^k \tag{16}$$

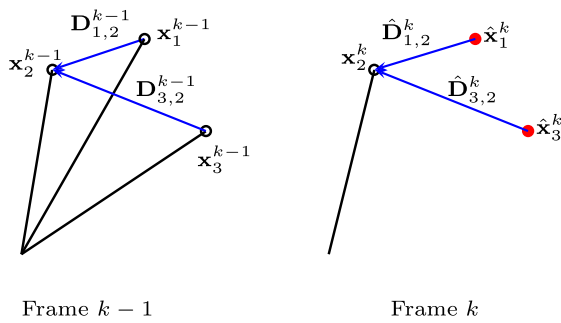


Fig. 4 The observation vector in the case of only one visible marker. The red dots, $\hat{\mathbf{x}}_j^k$, where $j = \{1, 3\}$ represent the estimated position of the missing marker m_j as given in (16)

where \mathbf{x}_2^k is the position of the visible marker m_2 on the limb in the current frame and $\hat{\mathbf{D}}_{j,2}^k$ is as described above. This assumes that there is no marker rotation. Figure 4 demonstrates the above.

5.4 All markers on a limb segment are missing

When all markers on a limb are occluded, we consider two possible subcases; the case where the other limb segment has some markers visible and the case where both limb segments have all of their markers occluded. If some markers on the other limb segment are visible (assume the y limb), the missing marker positions can be calculated using the CoR estimate, $\hat{\mathbf{C}}_k = \mathbf{y}_i^k - R_y^k \bar{\mathbf{a}}_y \tilde{R}_y^k$ where $i = \{1, 2, 3\}$. Our methodology takes advantage of the approach in [12] for CoR estimation and provides better approximations of the CoR using information from the adjacent limbs. The observation vector of the Unscented Kalman filter is then updated as

$$\mathbf{Z}_k^j = H \hat{\mathbf{x}}_j^k + \mathbf{v}_j^k \tag{17}$$

where $\hat{\mathbf{x}}_j^k$ is the estimated position of the occluded marker m_j ($j = 1, 2, 3$) in frame k . $\hat{\mathbf{x}}_j^k$ is given by

$$\hat{\mathbf{x}}_j^k = \hat{\mathbf{C}}_k + \hat{\mathbf{D}}_{j,c}^k \tag{18}$$

where $\hat{\mathbf{D}}_{j,c}^k$ is an estimate of the vector between marker m_j and the CoR. This approach takes advantage of the fact that the distance between markers and the CoR is constant. This vector is approximated by $\hat{\mathbf{D}}_{j,c}^k = R^{k-2,k-1} \mathbf{D}_{j,c}^{k-1} \tilde{R}^{k-2,k-1}$ where $\mathbf{D}_{j,c}^{k-1} = \mathbf{x}_j^{k-1} - \mathbf{C}_{k-1}$. This assumes that the rotation of the markers between two consecutive frames remains constant. Figure 5 illustrates this procedure.

If both limb segments have all markers occluded, only information from previous frames can be used. The observation vector, \mathbf{Z}_k^j , in this instance is calculated using a quaternion based method. This method also assumes that the segment rotation between two consecutive frames is constant.

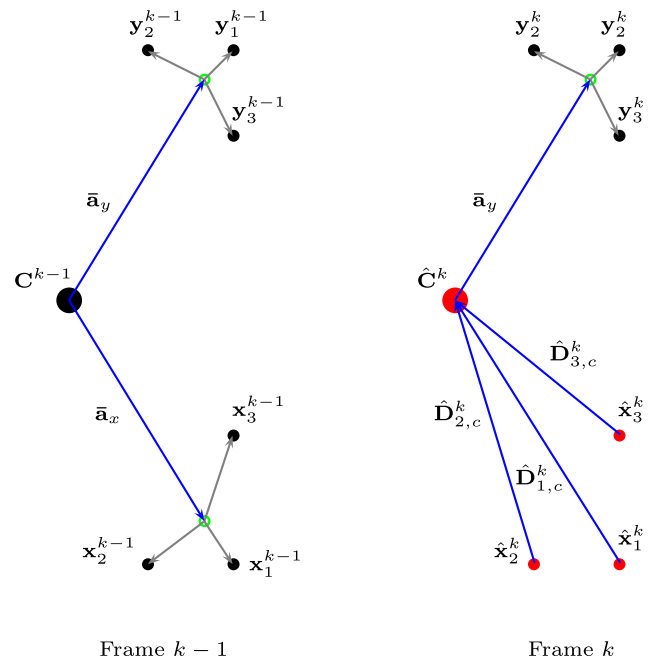


Fig. 5 The estimation procedure when all markers on a single limb segment are occluded. The red dots represent the estimated position of the CoR, $\hat{\mathbf{C}}^k = \mathbf{y}_i^k - R_y^k \bar{\mathbf{a}}_y \tilde{R}_y^k$ where $i = \{1, 2, 3\}$, and the estimated marker positions on limb segment x , $\hat{\mathbf{x}}_j^k = \hat{\mathbf{C}}_k + \hat{\mathbf{D}}_{j,c}^k$, where $j = \{1, 2, 3\}$. $\bar{\mathbf{a}}_x$ and $\bar{\mathbf{a}}_y$ are updated using the predicted marker positions in the current frame

The observation vector can now be expressed as

$$\mathbf{Z}_k^j = H \hat{\mathbf{x}}_j^k + \mathbf{v}_j^k \tag{19}$$

where $\hat{\mathbf{x}}_1^k$ is equal to $\hat{\mathbf{x}}_1^k = R^{k-2,k-1} \mathbf{x}_1^{k-1} \tilde{R}^{k-2,k-1}$. Assuming that the markers rotate constantly between two consecutive frames, the method performs better than using a simple variable velocity model since it ensures that markers do not move independently [4]. Markers on a limb segment move as a group of three and their inter-marker distance should remain constant over time. Note that we must ensure our rotor R satisfies $R\tilde{R} = 1$, and if this is not the case, normalisation must occur.

5.5 Markers visible in only one camera

Each marker can be reconstructed by the motion capture system if it is visible in at least two cameras. Indeed, looking at numerous real datasets, we have observed instances where the missing markers are not entirely occluded; information about position is often returned by a single camera. This information identifies a line, L_1 , starting from the camera and passing through the position of the missing marker. By relaxing the constraints that the inter-marker distance is constant and accepting that the real position of the marker is on that line, we may be able to obtain a more accurate estimate

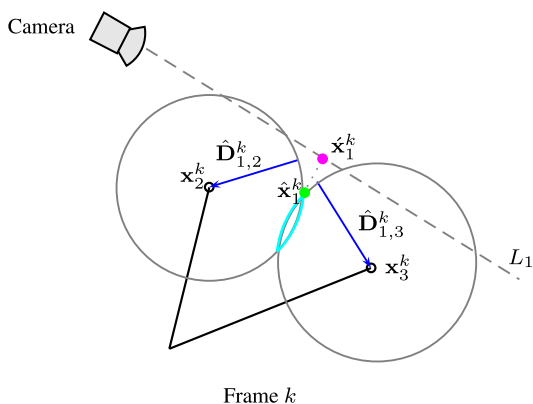


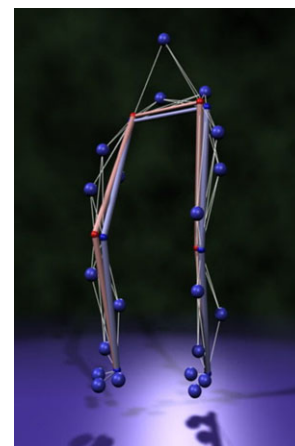
Fig. 6 The observation vector in the case of 2 visible markers and one marker visible only by one camera. The red dot, \hat{x}_1^k , is now used for the calculation of the observation vector, $Z_k^1 = H\hat{x}_1^k + v_1^k$

of the position of the relevant marker. This position, \hat{x}_1^k , corresponds to the projection from the point \hat{x}_1^k onto the line L_1 , as in Fig. 6. This is applicable for the cases in which the motion capture system fully reconstructs one or two marker locations and another marker is visible in just one camera. If a limb segment has only one known and one partially visible marker, the system is more reliable when it first predicts the partially visible marker and then the entirely occluded marker.

6 Bone length constraints using inverse kinematics

The UKF-VTM model for marker prediction returns good estimates of the marker positions, even if the markers are occluded for an extended period of time. It is, however, easily observable (see Fig. 7) that there are instances where the reconstructed markers may break the rigid body assumption (limbs may not have constant lengths over time), resulting in a violation of the model’s structure. This is more obvious in extreme cases where many markers from the same limb segment are occluded for an extended time period (see Fig. 15); the limb segment lengths drift (as the inter-joint distances were not fixed to be constant) even if the estimated marker positions do not significantly differ from their true positions. Therefore, it seems sensible to constrain the bone lengths to be constant over time, taking into account the rigidity of the body, as the skeleton methods do; this extension does not pre-suppose any knowledge of the model. For such unconstrained models we can preserve bone lengths using Inverse Kinematics techniques. Inverse Kinematics (IK) is defined as the problem of manipulating articulated figures in an interactive and intuitive fashion for the design and control of their posture. It was first introduced in the robotics area, but was then adapted in several other areas such as computer animation, ergonomics, and gaming. Ishigaki et al. [37] implement a real-time IK control interface for character ani-

Fig. 7 The results produced using the UKF-VTM model. It is seen that the distances between hips and knees change, since it is not guaranteed that these distances are constant. True positions are coloured blue, while the predicted positions are red



mation which translates the performance into corresponding actions; that was achieved by integrating prerecorded motions with online performance and dynamic simulation. If the input motion does not match the conditions, a kinematic process is applied to match users’ motion with the example interactions. However, the proposed inter-joint constraint approach requires offline training, meaning that the results depend on the training data, and prior knowledge of the model is required. To the best of our knowledge, this is the first time that a real-time IK technique has been used for CoR correction in the presence of missing elements in optical motion capture.

6.1 Related inverse kinematics work

Several models have been proposed for solving the IK problem from many different areas of study. Most papers use variants of the Jacobian Inverse approach. The *Jacobian solutions* are linear approximations of the IK problem; they linearly model the end effectors’ movements relative to instantaneous system changes in link translation and joint angle. Several different methodologies have been presented for calculating or approximating the Jacobian inverse, such as the Jacobian Transpose, Singular Value Decomposition (SVD), Damped Least Squares (DLS), Selectively Damped Least Squares (SDLS), and several extensions [7, 8, 11, 50, 65, 70]. Jacobian inverse solutions produce smooth postures; however, most of these approaches suffer from high computational cost, complex matrix calculations, and singularity problems. An alternative approach is given by Pechev in [53] where the problem is solved from a control perspective. This approach is computationally more efficient than the pseudo-inverse based methods and does not suffer from singularity problems.

Another family of IK solvers is based on Newton methods. These algorithms seek target configurations which are posed as solutions to a minimisation problem, hence they return smooth motion without erratic discontinuities. The most well known methods are Broyden’s method, Powell’s

method and the Broyden, Fletcher, Goldfarb and Shanno (BFGS) method [23]. However, the Newton methods are complex, difficult to implement and have high computational cost per iteration.

Courty and Arnaud, [15], proposed a Sequential Monte Carlo Method (SMCM) to incorporate kinematic constraints. Hecker et al., [29], utilised an iterative Inverse Kinematics solver (Particle IK) with various parameters for tuning the character skeleton behavior both statically and dynamically. Neither method suffers from matrix singularity problems and both perform reasonably well. Sumner et al., [62], and later Der et al., [17], proposed mesh-based IK solvers that learn the space of shapes from example meshes. However, this family of methods produces poses which require a pre-learning phase and are highly dependent on the training data.

A very popular IK solution is the Cyclic Coordinate Descent (CCD) method, first introduced by [67]. CCD is a heuristic iterative method with low computational cost for each joint per iteration, which can solve the IK problem without matrix manipulations. However, CCD can suffer from unrealistic animation, even if manipulator constraints have been added. It is designed to handle serial chains, thus it is difficult to extend to problems with multiple end effectors or target positions for internal joints.

A more detailed overview of IK techniques is given in [2]. Within this work, we incorporated FABRIK to solve the IK problem. FABRIK (Forward And Backward Reaching Inverse Kinematics) [3] is chosen since it is an iterative algorithm which uses points and lines to solve the IK problem. It divides the problem into 2 phases, a forward and backward reaching approach, and supports all the rotational joint limits and joint orientations by repositioning and re-orienting the target at each step. It does not suffer from singularity problems, it is fast and computationally efficient. FABRIK is simple, real-time implementable and can be easily extended to solve multiple end effector problems with internal joints.

6.2 Adjusting FABRIK for CoR correction

The tracking systems may return null data due to occlusions and that could cause interruptions in computation of pose and orientation of the target objects. The use of forecasted marker positions to compensate for these occlusion interruptions, and thus to drive real-time joint localisation, may introduce some error in the CoR estimation, breaking the rigid body assumption. Hence, inter-joint restrictions are incorporated using an adapted version of FABRIK for CoR correction.

6.2.1 Problems with serial chain models

FABRIK is a real-time IK solver which returns smooth postures in an iterative fashion. It uses the positions of the joints

that have been estimated using predicted marker positions to find updates that meet the fixed inter-joint distance assumption. In the most general case, where the estimated CoRs are not positioned at the end of the chain, the solution can be achieved using a forward and backward iterative mode. The proposed method starts from a joint of the chain which was calculated using true marker positions and works forward, adjusting each estimated joint along the way until the next joint which was calculated using true data. Thereafter, it works backward in the same way, in order to complete a full iteration. This method, instead of using angle rotations, treats finding the joint locations as a problem of finding a point on a line; hence, time and computation can be saved.

A graphical representation of the algorithm in action is given in Fig. 8. Assume $\mathbf{p}_1, \dots, \mathbf{p}_n$ are the joint positions of a chain, where \mathbf{p}_1 and \mathbf{p}_n are joint positions calculated using true data, and the joints inbetween are joints estimated using predicted marker positions. Set the distances between each joint to be $d_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|_{\text{avg}}$, for $i = 1, \dots, n - 1$, $\mathbf{b} = \mathbf{p}_1$ and $\mathbf{t} = \mathbf{p}_n$; these distances can be established by averaging over time from the frames where the markers, and the joints, are available. Then check whether the target is reachable or not; find the distance between \mathbf{p}_1 and \mathbf{p}_n , $dist$, and if this distance is smaller than the total sum of all the inter-joint distances, $dist < \sum_1^{n-1} d_i$, the target is within reach, otherwise, it is unreachable. If the target is within reach, a full iteration is performed in two stages. In the first stage, the algorithm estimates each joint position starting from \mathbf{p}_1 , moving forward to \mathbf{p}_n . Thus, initialise $\mathbf{p}_1 = \mathbf{b}$ and find the line, l_1 , which passes through the joint positions \mathbf{p}_1 and \mathbf{p}_2 . The new position of the 2nd joint, \mathbf{p}'_2 , lies on that line with distance d_1 from \mathbf{p}_1 . Similarly, the new position of the 3rd joint, \mathbf{p}'_3 , can be calculated using the line l_2 , which passes through \mathbf{p}_3 and \mathbf{p}'_2 , and has distance d_2 from \mathbf{p}'_2 . The algorithm continues until all new joint positions are calculated, including \mathbf{p}'_n .

Having in mind that initially \mathbf{p}_n was the true position of the n th joint, a second stage of the algorithm is needed. A full iteration is completed when the same procedure is repeated but this time starting from the true position of the n th joint and moving backward to the 1st joint. Therefore, let the new position for the n th joint, \mathbf{p}''_n , be its initial position \mathbf{t} . Then, using the line l_{n-1} that passes through the points \mathbf{p}''_n and \mathbf{p}'_{n-1} , we define the new position of the joint \mathbf{p}''_{n-1} as the point on that line with distance d_{n-1} from \mathbf{p}''_n . This procedure is repeated for all the remaining joints, including \mathbf{p}_1 . The procedure is then repeated, for as many iterations as needed, until \mathbf{p}_1 and \mathbf{p}_n are close enough (to be defined) to their initial, true positions. FABRIK always converges to any given chains/goal positions, when this is possible. If there are constraints which do not allow the chain to bend enough or if the target is not within the reachable area,

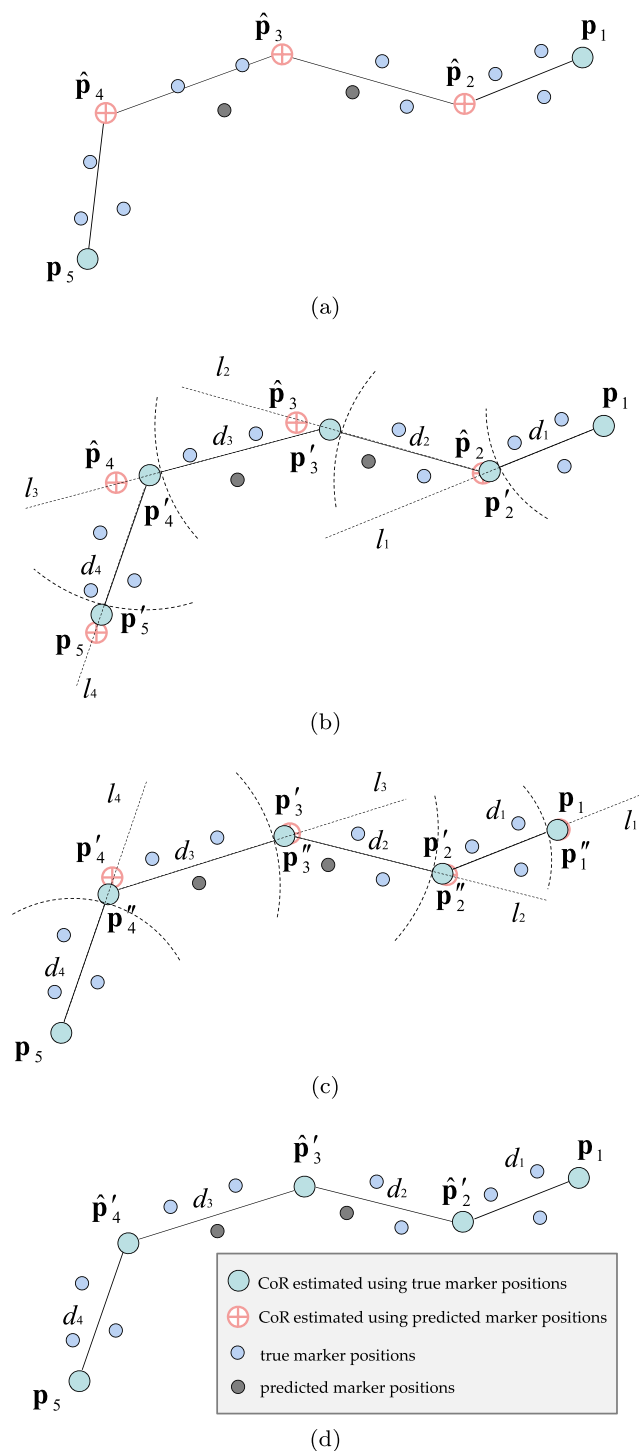


Fig. 8 A simple example of FABRIK for the case where the estimated joints are positioned inbetween 2 true joint positions. (a) The initial position of the chain, where $\{p_i\}$ are true and $\{\hat{p}_i\}$ are estimated joint positions. (b) The first phase of the algorithm; the joint \hat{p}_i has been adjusted as the point on line l_{i-1} with distance d_{i-1} from p_{i-1} . (c) The second phase of the algorithm: let the new position of p'_5 be its initial position p_5 ; repeat the procedure starting this time from the other side of the chain. (d) The final posture; the algorithm is repeated for as many iterations as needed until the difference between the true and the returned positions of the joints p_1 and p_5 is less than a given tolerance

there is a termination condition which compares the previous and the current position of the joint p_1 , and if this distance is less than a specified tolerance, FABRIK terminates its operation. Several optimisations can be achieved using Conformal Geometric Algebra (CGA) [19, 32] to produce faster results and to converge to the final answer in fewer iterations; CGA has the advantage that basic entities, such as spheres, planes, or circles, are simply represented by algebraic objects. Therefore, a direct estimate of a missing joint, when it is between 2 true positions, can be achieved by intersecting 2 spheres with centre the true joint positions and radii the distances between the estimated and the true joints, respectively. Another simple optimisation is the direct construction of a line pointing toward p_n , when the target is unreachable, adjusting the distances in such a way that each distance has changed uniformly. The adjusted FABRIK for CoR correction is illustrated in pseudo-code in Algorithm 1.

There are, indeed, some special instances where FABRIK does not necessarily work in an iterative fashion. Such a special case is when the joints, which have been estimated using predicted markers, are located at the end of the chain. This is a simplified case of the general solution, since the answer is given directly in one single iteration. The new simplified algorithm is given here: assume p_i is a true joint position. The joint update p'_{i+1} is set as the point on line l_i that passes through p_i and p_{i+1} and has d_i distance from p_i . This procedure is repeated for all the remaining estimated joints until the end of the chain. A graphical representation of an implemented example is given in Fig. 9.

6.2.2 Problems with tree models

In reality, most of the real legged models are comprised of several kinematic chains in a tree fashion. The proposed algorithm can be easily extended to process tree models with multiple serial chains. The solution is similar to the original extension of FABRIK for multiple end effectors, given in [3]. Hence, if the sub-base³ joint is a true position, FABRIK is used on each chain with estimated CoR positions individually, starting from the sub-base, in a forward and backward iterative fashion. If the sub-base is an estimate, the same procedure is applied but this time starting from the true joints of the connected chains and moving towards to the sub-base. This will produce as many different positions of the sub-base as the number of the connected chains. The new position of the sub-base will then be the centroid of all these positions. In the second stage, the normal algorithm is applied separately for each chain, starting now from the sub-base and moving outward to the starting joints. The method is repeated until all true joints have no significant change between their initial and updated positions.

³A sub-base joint is a joint which connects 2 or more chains.

Algorithm 1: A full iteration of FABRIK for CoR correction

Input: The joint positions \mathbf{p}_i for $i = 1, \dots, n$, and the average distances between each joint $d_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|_{\text{avg}}$ where n is the number of joints of the chain.

Output: The new joint positions \mathbf{p}_i for $i = 1, \dots, n$.

```

0.1  $\mathbf{b} = \mathbf{p}_1$ ;  $\mathbf{t} = \mathbf{p}_n$ ;
0.2 % Initialisation of the difA value.
0.3 difA = 1
0.4 while difA > tol do
0.5     % STAGE 1: FORWARD REACHING
0.6     % Set  $\mathbf{p}_1$  to its initial position.
0.7      $\mathbf{p}_1 = \mathbf{b}$ 
0.8     for  $i = 1, \dots, n - 1$  do
0.9         % Find the distance  $r_i$  between the estimated joint
10         % position  $\mathbf{p}_{i+1}$  and the joint  $\mathbf{p}_i$ 
11          $r_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|$ 
12          $\lambda_i = d_i / r_i$ 
13         % Find the new joint positions  $\mathbf{p}_i$ .
14          $\mathbf{p}_{i+1} = (1 - \lambda_i)\mathbf{p}_i + \lambda_i\mathbf{p}_{i+1}$ 
15     end
16     % STAGE 2: BACKWARD REACHING
17     % Set  $\mathbf{p}_n$  to its initial position.
18      $\mathbf{p}_n = \mathbf{t}$ 
19     for  $i = n - 1, \dots, 1$  do
20         % Find the distance  $r_i$  between the joint position
21         %  $\mathbf{p}_{i+1}$  and the estimated joint  $\mathbf{p}_i$ 
22          $r_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|$ 
23          $\lambda_i = d_i / r_i$ 
24         % Find the new joint positions  $\mathbf{p}_i$ .
25          $\mathbf{p}_i = (1 - \lambda_i)\mathbf{p}_{i+1} + \lambda_i\mathbf{p}_i$ 
26     end
27 difA =  $|\mathbf{p}_1 - \mathbf{b}|$ 
28 end

```

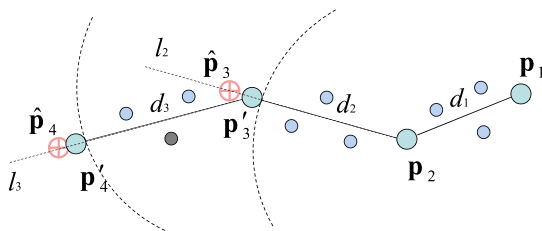


Fig. 9 A simple case where the estimated joints are located at the end of the chain. In this example, the serial chain has 4 joints where \mathbf{p}_1 and \mathbf{p}_2 have been calculated using true marker positions and the $\hat{\mathbf{p}}_3$ and $\hat{\mathbf{p}}_4$ using predicted markers positions. Thus, set \mathbf{p}'_3 to be the point on line l_2 which has distance d_2 from \mathbf{p}_2 and \mathbf{p}'_4 the point on line l_3 that has d_3 distance from joint \mathbf{p}'_3

6.2.3 Applying constraints

The main aim of this paper is to describe a general solution, where each single joint is treated as a ball and socket joint, allowing 3 degrees of freedom. In this way, we can ensure the generality of the proposed method, producing solutions without prior knowledge of the model. The proposed IK technique is used just to correct mis-positioning of joint estimates in the presence of missing data. The orientation of each joint is still calculated using the attached sensor LEDs (or their forecasted positions). In special cases where it is desirable to incorporate joint restrictions, especially for cases where joints are located at the end of the kinematic chains, FABRIK readjusts the target position and orientation, on each step, to satisfy the joint biomechanical limits. Note that FABRIK is able to support most of the joint types, as detailed in [3].

Obviously, the more information available regarding the model's structure and joint constraints, the more accurate and efficient will be the results. This information can help to give a visually more realistic motion within a feasible set; however, it will limit the universal use of the proposed methodology and fails to satisfy our aim of no prior knowledge of the model.

Self-collision determination Collision detection has been a fundamental problem in computer animation, physically-based modelling, geometric modelling, and robotics. Since the data used in these examples are captured from a marked optical motion capture system and since the 3d animated humanoids do not have a mesh that defines their external shape, self-collisions are not considered. Nevertheless, self-collisions can be handled using existing techniques, such as [10, 44].

Bone length constraints The Inverse Kinematic technique used in this work ensures that, even if the marker prediction system fails to track the marker positions, the derived CoRs will be good estimates of their true positions. This happens since the IK procedure restricts the limb segment (bone) length to a constant value over time. The true length of each limb segment is calculated by averaging the inter-joint distances, from previous frames, when all joints have been estimated using true marker positions. Figure 10 shows an example of how the inter-joint distance varies over time; as expected, it is nearly constant. The predicted marker positions can cause instability in the inter-joint distance, which in some extreme cases might differ significantly from the mean bone length. The effectiveness of the proposed method is strongly related to the stability of the joint pairwise distance. Large variation of that distance means skeletal structure violation. In this manner, FABRIK guarantees that joints will still have a fixed distance between them throughout the occlusion period, allowing a better approximation of the CoR

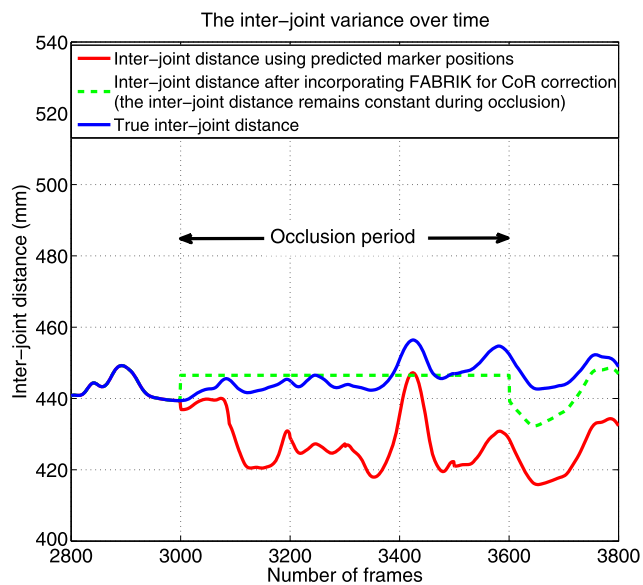


Fig. 10 An example showing the inter-joint distance variation over time when true marker positions have been used, when the joints have been estimated using predicted marker positions, and after incorporating FABRIK for CoR correction (assuming that the distance is constant over time)

estimates. On real motion capture data, limb lengths might exhibit minor variations due to noise from skin movements at the point of marker attachment, and limb expansions, but in this work they are assumed to be rigid. Note that the limb length variation is further reduced when the data have been isolated using the proposed UKF-VTM method, as is briefly described in Sect. 7.

It is important to stress out that the proposed bone length constraint algorithm (FABRIK) is independent and can be easily fitted to most marker prediction methods for CoR correction.

7 Results and discussion

Experiments were carried out using a 24 camera PhaseSpace motion capture system capable of capturing data at 480 Hz [54]. The algorithm described in this work can process up to 300 *limb segment pairs per second* (using MATLAB). Our datasets comprise real data (i.e. captured data with natural occlusions or occlusions generated by artificial deletion) with more than 10,000 *frames* in each. It is technically impossible to evaluate the error when a benchmark does not exist to compare with. Thus, different numbers of markers per limb segment have been deleted randomly, in different parts of the human body, to enable error calculation. Nevertheless, the estimates of the missing markers and CoRs in instances where natural deletions exist have been evaluated visually based on the assumption that the character displays smooth and natural motion. The proposed

methodology has been tested on various motions including dancing (14 segment body datasets), fast walking (7 segment leg datasets) and boxing (5 segment arm datasets). Using real data with occlusions generated by artificial deletion, we are able to calculate the error of the proposed methodologies; the error is measured as the average distance between the true and the estimated position (for marker and the CoRs) after many runs and different artificial deletions. As expected, the error varies between different instances of marker occlusion. As more markers become available, more information relative to the limb segment is available and thus, a higher accuracy is achieved. The magnitude of the error reported in the results is given in terms of real world distances. Within this work, our methodology (referred as UKF-VTM) was tested and compared against some of the most popular marker prediction approaches: an EKF model, such as [1], using similar marker constraints to the ones proposed here, a cubic spline interpolation and a real-time extrapolation method with marker position constraints, as reported in [55].

Using the VTM we took into consideration the velocity, direction, and acceleration changes of the markers' trajectories over time. In addition, we considered the rotation of the limbs over time for a more accurate estimation of the marker positions. Certain drawbacks of the simple EKF were overcome by using a more evolved and sophisticated method; the use of a Variable Turn Model gives significant improvements in cases where the trajectories of the markers are variable and have abrupt fluctuations in speed and direction. The proposed system (UKF-VTM), without applying FABRIK for bone constraints, returns an average error (over 20 runs) of 1.296 cm in the case of one missing marker, 3.474 cm when 2 markers are missing, and 8.401 cm when all markers are occluded. The corresponding CoR estimation error is 1.082 cm in the case of one missing marker, 1.9867 cm in the case of two markers and 7.859 cm in the case where no markers are visible to the cameras. Table 1 lists and compares the results of each method implemented here under several occlusion scenarios. In general, the methods fail to track the missing marker paths when the marker constraints were not incorporated, especially in instances where the occlusion lasts longer than a time threshold. Table 1 also tabulates the worst case scenario (distance between true and predicted positions) of marker prediction and CoR estimation in addition to the standard deviation. The worst case scenarios usually appear at abrupt changes in velocity and direction of the missing marker during the occlusion period, where the UKF-VTM model requires some time to efficiently track the target.

Obviously, the interpolation method has the lowest computation cost, but it is an off-line application. The UKF-VTM framework increased the processing time by 20% compared to the simple EKF, processing 315 *limb segment*

Table 1 Average results (over 20 runs) under multiple cases of occlusion (large occlusions of 2,500 frames in total)

Method	Missing markers on a limb segment	Markers			CoRs			Proc. limbs per second
		Error (cm)	Worst case scenario (cm)	Standard deviation (cm)	Error (cm)	Worst case scenario (cm)	Standard deviation (cm)	
UKF-VTM	One marker	1.2958	3.5677	1.3412	1.0820	1.7888	0.5897	315
	Two markers	3.4737	7.6584	2.5149	1.9867	2.7845	0.5904	
	All markers	8.4012	12.8749	3.8787	7.8591	13.1183	1.1499	
Interpolation	One marker	9.4687	15.2552	5.1709	6.1189	15.8777	5.1825	900
	Two markers	10.5241	16.8874	5.8501	8.5874	17.3340	5.9889	
	All markers	11.0010	17.2282	5.9918	12.2514	20.2111	6.7001	
Extrapolation	One marker	1.8574	4.8811	2.4167	1.5874	3.1147	1.2036	270
	Two markers	4.5214	9.0444	3.9057	2.5147	3.5558	0.5587	
	All markers	10.5961	13.8541	2.4808	11.5824	15.2588	2.7193	
EKF	One marker	1.3982	3.8971	2.1071	1.2356	2.1496	0.6654	370
	Two markers	3.5331	8.0145	2.7060	2.0302	3.0098	0.8944	
	All markers	8.4205	13.0014	3.0182	7.9055	13.1198	3.0018	

Table 2 Average results (over 20 runs) on real data with occlusions generated by deletions. Case of one missing marker on each limb segment for more than 1500 frames.

	Entirely occluded (error cm)	Partially visible (error cm)	Change
Markers	1.3458	0.2554	−81.02%
CoRs ^a	2.2247	0.5628	−74.70%

^awhen \bar{a}_w is updated using the predicted data

pairs per second in MATLAB; which does, however, still allow real-time implementation. The UKF-VTM method performs better than the other methods looked at here, resulting in the most accurate results; our methodology efficiently recovers good estimates of the missing markers and accurate real-time CoR estimation.

A further error reduction was observed when the missing markers were partially visible to one camera. The error is significantly decreased, by 80% for marker prediction and 75% for CoR estimation, compared to the case where this information was ignored. Table 2 gives the prediction error for the case of one missing marker on each limb segment when the missing markers are both entirely occluded and visible in just one camera.

The difference between true and estimated positions is further reduced when the fixed inter-joint pairwise assumption was taken into consideration. The CoR error, in our methodology, is decreased on average (over 30 runs) by 11.9% in total, 2.96% when the estimated joints are located at the end of the chain and 12.7% when they are positioned between 2 true positions. FABRIK ensures that the inter-joint pairwise distances are constant over time, thus elim-

inating the error in the CoR estimation due to unnatural bone extensions. The error reduction is larger in cases where the estimated joints are between 2 true CoR positions, since both true joints, in an iterative fashion, constrain the inter-joint distance from the estimated joints. In the case where the estimated joints are positioned at the end of the chain, only one true joint contributes in the final solution. Table 3 presents the average error and achieved error reduction after applying IK to the case of a humanoid model with artificial deletions on 6 markers (over 30 markers in total) for more than 2,500 frames out of a total of 8,000. At the same time, the processing time was increased by only 5.27%, processing now approximately 300 *limb segment pairs per second*. FABRIK has been applied to all methodologies ensuring that the inter-joint distances will remain unchanged over time; the CoR error was reduced on average by 12.92% for the EKF, 45.33% for the extrapolation method, and 62.8% for the interpolation approach.

Figure 11(a) shows examples of the true and predicted x -positions of an occluded marker over time, and its corresponding CoR (after incorporating FABRIK for CoR correction to all methods) for the case of a single occlusion. It is clear that the occluded marker can be tracked with high accuracy when it is visible in at least one camera and its CoR position can be reconstructed efficiently even if the occlusion period exceeds 1,000 frames. Note that the joint estimates differ from their true positions, even after the occlusion period, since the rotor R^k expressing the rotation between the first and the current frame is an estimation using the predicted marker positions (see (4) and (5)). Nevertheless, the error of the CoR is reduced over time (when all markers are visible) because the new markers positions

Table 3 Error reduction using Inverse Kinematics

Method		Overall results		Joints are located at the end of the chain		Joints are located in between 2 true joint positions	
		Error (cm)	Change	Error (cm)	Change	Error (cm)	Change
UKF-VTM	CoR using only predicted marker positions	2.0268		2.1013		2.0131	
	Corrected CoR using FABRIK	1.7856	-11.90%	2.0391	-2.96%	1.7574	-12.70%
Interpolation	CoR using only predicted marker positions	7.7149		8.3149		7.2650	
	Corrected CoR using FABRIK	2.8698	-62.80%	3.7950	-54.35%	2.2937	-68.42%
Extrapolation	CoR using only predicted marker positions	4.2437		4.9334		3.7263	
	Corrected CoR using FABRIK	2.3200	-45.33%	2.8495	-42.25%	1.9712	-47.10%
EKF	CoR using only predicted marker positions	2.5077		2.8684		2.3509	
	Corrected CoR using FABRIK	2.1836	-12.92%	2.5474	-11.19%	2.0212	-14.02%

give a more accurate estimate of the rotor. Figure 11(b) also shows an example of the error variation for all methodologies due to occlusion for the case of 1 missing marker on each limb segment. Clearly, the hole-fitting achieved using the UKF-VTM method has the lowest error for both the average and the worst case scenario; this error is further decreased when the marker is visible to just one camera.

Figure 12 reinforces the above showing the cumulative distribution function (CDF) of the estimation error for the case of one missing marker on each limb segment. The x -axis shows the estimation error and the y -axis shows the probability, for $y = a$, of having an error less than or equal to a . Hence, for example, the probability that the estimation error is less than or equal to 1.2 cm is approximately 0.57 for the extrapolation and less than 0.4 for interpolation, while it is 0.62 for the case of the UKF-VTM. The median estimation error of the markers using the EKF model is approximately 1.12 cm (0.83 cm for the CoR), where the corresponding median error for the case of UKF-VTM is approximately 1.09 cm (0.75 cm for the CoR and 0.12 cm when markers are visible to just one camera).

The interpolation method produces smooth results since it uses previous and future positions. Our method, along with other approaches that use data only from previous frames, exhibits small oscillations or discontinuities between the last predicted position and the first frame when the marker again becomes visible. This is expected as no future information is used. We can avoid this phenomenon if, instead of using the true marker position as the final result, we continue to use the UKF framework. This way we will have smoother results with just a marginal increase in the marker and CoR errors.

Extrapolation and interpolation return useful predictions for short-time occlusions but fail to track the marker positions when the occlusion is maintained for extended time periods, especially if markers change rapidly in direction and

velocity. In particular, the path followed by the interpolation method does not reflect the actual state of the missing marker, but only connects the available positions using a cubic spline shape. In contrast, the UKF-VTM performs fairly well even if markers are missing for long time periods.

Figure 13 shows an example of using our algorithm on real data; in blue are the true positions of the markers and CoRs and in red the predicted positions. Figure 14 is another example that compares the results on the leg model; (a) presents the results when only the integrated UKF-VTM model is used, and (b) shows the results when bone length control was incorporated using FABRIK. Note the bone length violation in Fig. 14(a), which is most obvious on the hips.

Figure 15 shows another example of implementation under an extreme case with extended marker occlusions. The top picture of the figure shows the results when the fixed inter-joint distance was not imposed and the bottom picture when FABRIK was applied. Obviously, in the first case the skeletal structure was violated since the bones were not restricted to their original lengths; in the second case the results have been improved to a visually more natural shape.

Experiments demonstrate that the proposed methodology can effectively track the occluded markers with high accuracy, even if they are occluded for extended periods of time, recovering in real-time good estimates of the true joint positions. FABRIK controls and corrects the CoR estimates decreasing the error between the estimated and true positions, thereby enabling real-time skeletal reconstruction. The resulting motion is natural and smooth, without oscillations and discontinuities, resembling that of true human movements.

The proposed approach, as well as predicting the coordinates of the invisible LEDs to prevent intermittent tracking, can also be used to isolate noisy data or bad inputs from in-

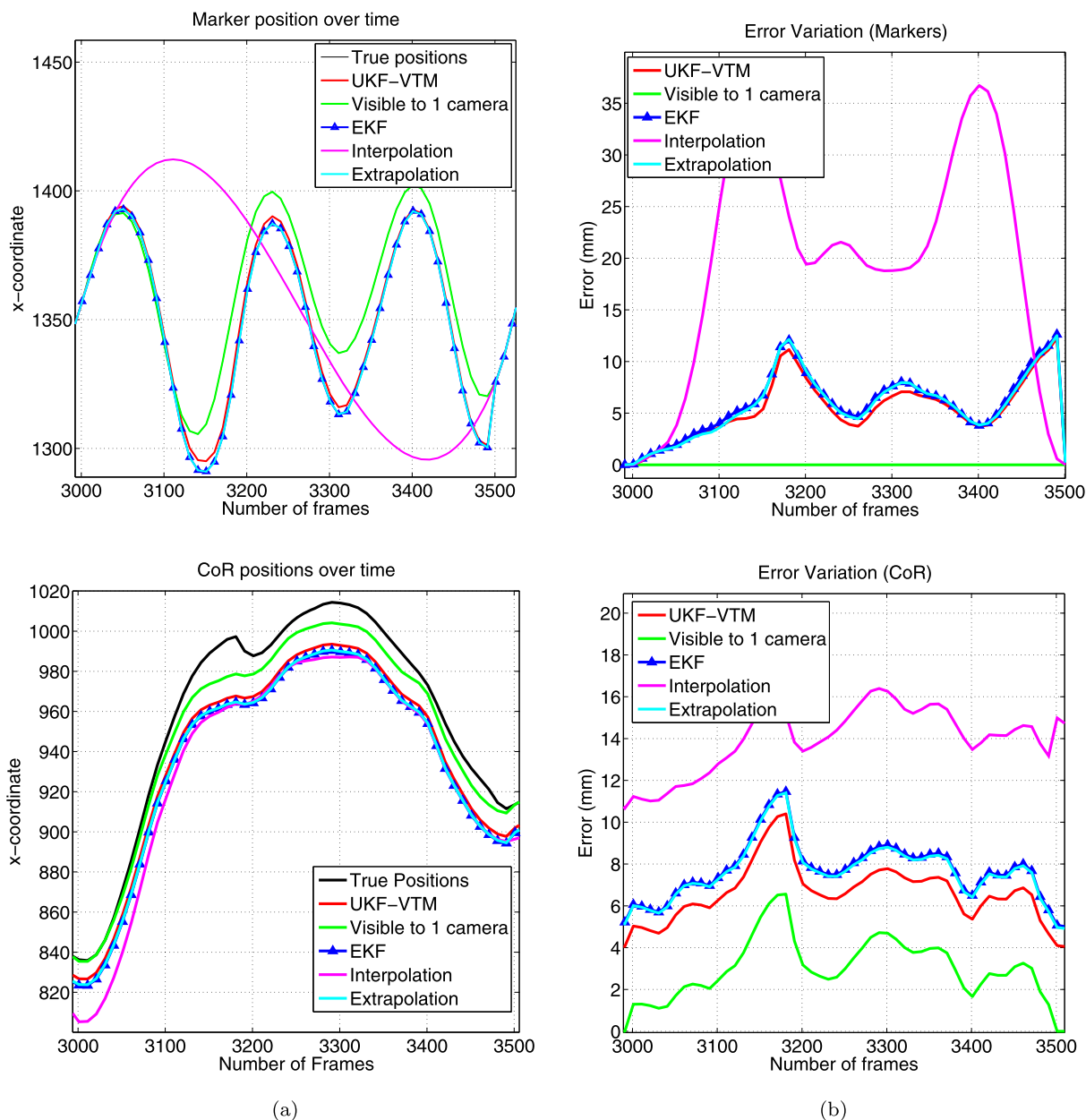


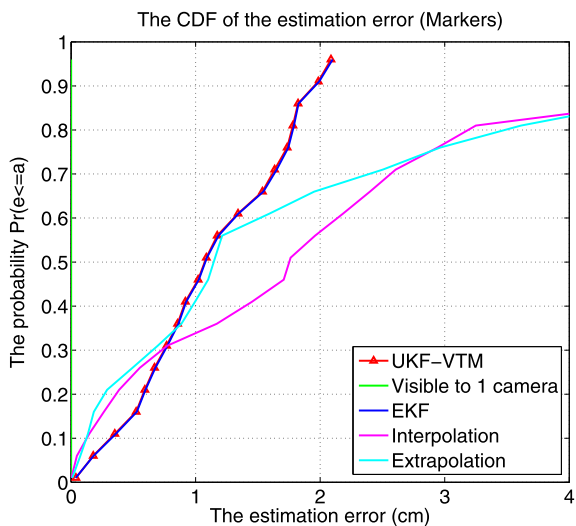
Fig. 11 Example showing comparison results for each methodology for the case of 1 missing marker on each limb segment; occlusions were introduced to this dataset for extended time periods of up to 1,500 consecutive frames. The graph shows the forecasted trajectories when the occlusion was between frames 3,000 and 3,600. The *upper row* shows

the results for marker predictions and the *lower row* for CoR estimates, after applying FABRIK. **(a)** The zoomed *x*-coordinate path over time (in this example note that the *black* and *green* curves for the markers are exactly overlaid), **(b)** the corresponding error variation over time

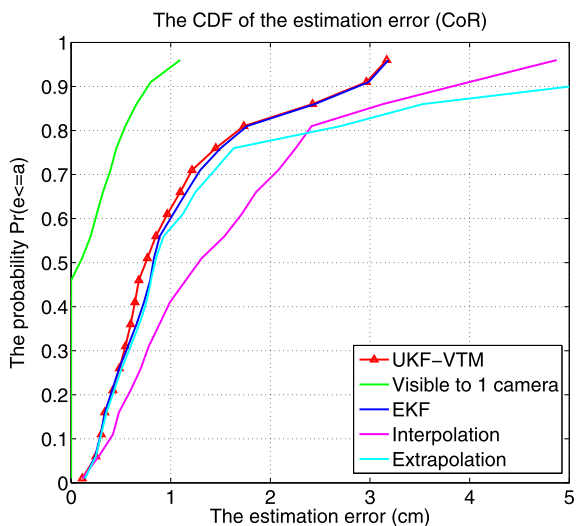
dividual motion capture cameras. As discussed in Sect. 4, motion capture data are noisy due to optical measurement problems, mis-calibration of the optical systems, skin movements, etc. After isolating the data using our UKF-VTM approach, smoother trajectories for both the marker and CoR coordinates has been perceived. It is also observed that the bone lengths are more constant, reducing the standard deviation from 4.2063 cm to 3.9185 cm, when the isolated data have been used.

8 Conclusion

This paper considers the problem of forecasting the occluded marker positions of optical motion capture systems to prevent intermittent tracking and automatically establish a skeleton model to which the markers are attached. An Unscented Kalman Filter framework was deployed for marker prediction using a Variable Turn Model as the observation sequence, in addition to a constant rotation assumption over



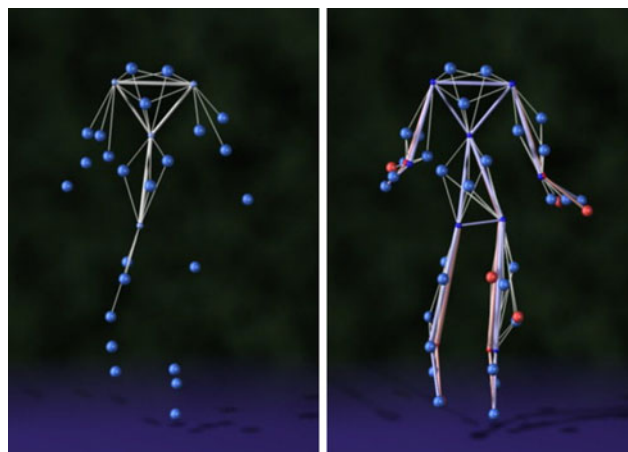
(a)



(b)

Fig. 12 Example showing the CDF of the estimation error (over 10 runs) for each methodology for the case of 1 missing marker on each limb segment: (a) the markers CDF and (b) the CoR CDF

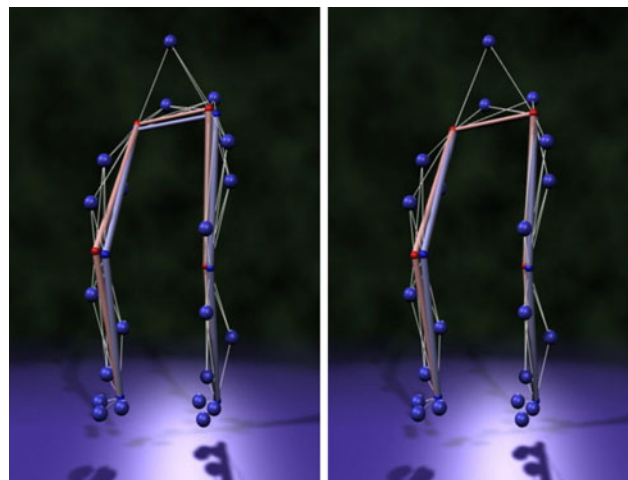
time. Inferred information from neighbouring markers has been used to give observation states via an approximate rigid body assumption. The proposed marker constraint model is simple and real-time implementable. At the same time, the proposed system is the first method which takes advantage of the information returned by each single camera. The predicted data is used for real-time joint localisation and the joint positions are then re-positioned using an Inverse Kinematics technique by taking into account the fact that inter-joint distances are constant over time. The work is novel in that it incorporates an adjusted version of FABRIK, a real-time iterative IK solver, which ensures fixed bone lengths over time, thus resulting in a more feasible mo-



(a)

(b)

Fig. 13 An example of marker prediction and CoR estimation using the proposed methodology. The true positions are coloured *blue* and the predicted positions *red*. (a) The artificially deleted data, (b) the predicted data



(a)

(b)

Fig. 14 An example of implementation. *Blue* represents the true positions and *red* the predicted positions: (a) using only the integrated UKF-VTM framework, (b) using FABRIK for bone length control

tion. Our methodology outperforms in accuracy the methods used for comparison in this work. It is able to maintain real-time marker predictions, thus enabling good estimates of the CoR, even in the presence of several marker occlusions on each limb segment. It is reliable in datasets with large sequences of null data, even if the limb rapidly changes velocity and direction. The proposed technique could also be used to isolate bad inputs from single cameras; it is a common phenomenon that even one error in a large number of marker inputs can result in several errors per second, pulling the marker position out of the expected path. Thus, this method can be utilised to eliminate pops and jumps from camera

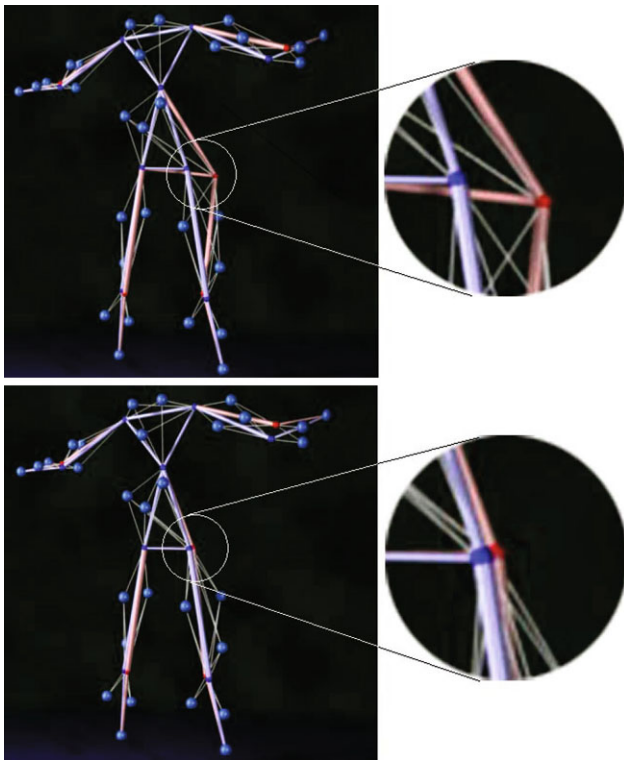


Fig. 15 An example of implementation under extreme cases with extended data occlusion. The *top picture* shows results using the integrated UKF-VTM; the *lower picture* shows the results when FABRIK was applied in order to maintain the fixed inter-joint distance assumption. The true positions are coloured in *blue* and the predicted in *red*

switching and errors as they fall outside the predicted positions.

Future work will introduce biomechanical constraints to restrict motions to those from a feasible set; however, prior knowledge of the model and joints will then be needed. Also, a hybrid system with low cost inertial measurement units could be used to validate the method proposed here.

References

1. Aristidou, A., Cameron, J., Lasenby, J.: Predicting missing markers to drive real-time centre of rotation estimation. In: Proceedings of the V Conference on Articulated Motion and Deformable Objects, Mallorca, Spain. LNCS, vol. 5098, pp. 238–247 (2008)
2. Aristidou, A., Lasenby, J.: Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver. Tech. Rep. F-INFENG/TR. 632, CUED (2009)
3. Aristidou, A., Lasenby, J.: FABRIK: a fast, iterative solver for the inverse kinematics problem. *Graphical Models* **73**(5), 243–260 (2011)
4. Aristidou, A., Lasenby, J., Cameron, J.: Methods for real-time restoration and estimation in optical motion capture. Tech. Rep. F-INFENG/TR. 619, CUED (2009)
5. Asseo, S.J., Ardila, R.J.: Sensor independent target state estimator design and evaluation. In: Proceedings of the National Aerospace and Electronics Conference (NAECON), pp. 916–924 (1982)
6. Backer, A.S.: Estimating missing motion capture data with accelerometers. MPhil thesis, Cambridge University Engineering Department, Cambridge, UK (August 2009)
7. Baillieul, J.: Kinematic programming alternatives for redundant manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 722–728 (1985)
8. Balestrino, A., De Maria, G., Sciavicco, L.: Robust control of robotic manipulators. In: Proceedings of the 9th IFAC World Congress, vol. 5, pp. 2435–2440 (1984)
9. Broeren, J., Sunnerhagen, K.S., Rydmark, M.: A kinematic analysis of a haptic handheld stylus in a virtual environment: a study in healthy subjects. *Journal of NeuroEngineering and Rehabilitation* **4**, 13 (2007)
10. Brown, J., Latombe, J.C., Montgomery, K.: Real-time knot-tying simulation. *The Visual Computer* **20**(2), 165–179 (2004)
11. Buss, S.R., Kim, J.S.: Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools* **10**(3), 37–49 (2005)
12. Cameron, J., Lasenby, J.: A real-time sequential algorithm for human joint localization. In: ACM SIGGRAPH Posters, p. 107. ACM, New York (2005)
13. Chai, J., Hodgins, J.K.: Performance animation from low-dimensional control signals. *ACM Transactions on Graphics (TOG)* **24**(3), 686–696 (2005)
14. Chang, L.Y., Pollard, N.: Constrained least-squares optimization for robust estimation of center of rotation. *Journal of Biomechanics* **40**(1), 1392–1400 (2007)
15. Courty, N., Arnaud, E.: Inverse kinematics using sequential Monte Carlo methods. In: Proceedings of the V Conference on Articulated Motion and Deformable Objects, Mallorca, Spain. LNCS, vol. 5098, pp. 1–10 (2008)
16. Courty, N., Cuzol, A.: Conditional stochastic simulation for character animation. *Computer Animation and Virtual Worlds—CASA' 2010* **21**(3–4), 443–452 (2010)
17. Der, K.G., Sumner, R.W., Popović, J.: Inverse kinematics for reduced deformable models. In: ACM SIGGRAPH Papers, pp. 1174–1179. ACM, New York (2006)
18. Dessai, S.S., Hornung, A., Kobbelt, L.: Automatic data normalization and parameterization for optical motion tracking. *J. Virtual Real. Broadcast.* **3**(3) (2006)
19. Doran, C., Lasenby, A.: *Geometric Algebra for Physicists*. Cambridge University Press, Cambridge (2003)
20. Dorfmueller-Ulhaas, K.: Robust optical user motion tracking using a Kalman filter. Tech. Rep. TR-2003-6, Institut fuer Informatik, Universitätsstr. 2, 86159 Augsburg (2003)
21. Doucet, A., De Freitas, N., Gordon, N.: *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
22. Ehrig, R.M., Taylor, W.R., Duda, G.N., Heller, M.O.: A survey of formal methods for determining the centre of rotation of ball joints. *Journal of Biomechanics* **39**(15), 2798–2809 (2006)
23. Fletcher, R.: *Practical Methods of Optimization*, 2nd edn. Wiley-Interscience, New York (1987)
24. Gamage, S.S.H.U., Lasenby, J.: New least squares solutions for estimating the average centre of rotation and the axis of rotation. *Journal of Biomechanics* **35**(1), 87–93 (2002)
25. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. In: ACM Transactions on Graphics (TOG), pp. 522–531. ACM, New York (2004)
26. Halvorsen, K.: Bias compensated least squares estimate of the center of rotation. *Journal of Biomechanics* **36**, 999–1008 (2003)
27. Halvorsen, K., Lesser, M., Lundberg, A.: A new method for estimating the axis of rotation and the center of rotation. *Journal of Biomechanics* **32**, 1221–1227 (1999)
28. Hashiguchi, J., Nivomiya, H., Tanaka, H., Nakamura, M., Nobuhara, K.: Biomechanical analysis of a golf swing using motion capture system. In: Proceedings of Annual Meeting of Japanese Society for Orthopaedic Biomechanics, vol. 27, pp. 325–330 (2006)

29. Hecker, C., Raabe, B., Enslow, R.W., Dewese, J., Maynard, J., van Prooijen, K.: Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)* **27**(3), 1–11 (2008)
30. Herda, L., Fua, P., Plänklers, R., Boulic, R., Thalmann, D.: Skeleton-based motion capture for robust reconstruction of human motion. In: *Proceedings of the IEEE Computer Animation (CA'00)*, pp. 77–86 (2000)
31. Herda, L., Fua, P., Plänklers, R., Boulic, R., Thalmann, D.: Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science Journal* **20**(3), 313–341 (2001)
32. Hestenes, D., Sobczyk, G.: *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Reidel, Dordrecht (1984)
33. Holzreiter, S.: Calculation of the instantaneous centre of rotation for a rigid body. *Journal of Biomechanics* **24**(7), 643–647 (1991)
34. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* **4**, 629–642 (1987)
35. Hornung, A., Sar-Dessai, S.: Self-calibrating optical motion tracking for articulated bodies. In: *Proceedings of the IEEE Conference on Virtual Reality, VR'05*, pp. 75–82. IEEE Computer Society, Washington (2005)
36. Hsu, E., Gentry, S., Popović, J.: Example-based control of human motion. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'04)*, pp. 69–77. Eurographics Association, Grenoble (2004)
37. Ishigaki, S., White, T., Zordan, V.B., Liu, C.K.: Performance-based control interface for character animation. *ACM Transaction on Graphics (TOG)* **28**(3), 1–8 (2009)
38. Jazwinski, A.H.: *Stochastic Processes and Filtering Theory*. Academic Press, San Diego (1970)
39. Julier, S.J., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, vol. Acquisition, Tracking and Pointing XI, Florida, USA, pp. 182–193 (1997)
40. Kalman, R.E.: A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Eng.*, 35–45 (1960)
41. Li, L., McCann, J., Pollard, N.S., Faloutsos, C.: Dynammo: mining and summarization of coevolving sequences with missing values. In: *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, pp. 507–516. ACM, Paris (2009)
42. Li, L., McCann, J., Pollard, N.S., Faloutsos, C.: Bolero: a principled technique for including bone length constraints in motion capture occlusion filling. In: *Proceedings of the ACM Symposium on Computer Animation*, Madrid, Spain (2010)
43. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking. Part I: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems* **39**(4), 1333–1364 (2003)
44. Lin, M.C., Gottschalk, S.: Collision detection between geometric models: a survey. In: *Proceedings of IMA Conference on Mathematics of Surfaces*, pp. 37–56 (1998)
45. Liu, G., McMillan, L.: Estimation of missing markers in human motion capture. *The Visual Computer* **22**(9–11), 721–728 (2006)
46. Liu, G., Zhang, J., Wang, W., McMillan, L.: Human motion estimation from a reduced marker set. In: *I3D'06: Proceedings of the Symposium on Interactive 3D Graphics and Games*, pp. 35–42. ACM, New York (2006)
47. Maidi, M., Ababsa, F., Malle, M.: Handling occlusions for robust augmented reality systems. *EURASIP J. Image Video Process.* **2010** (2010)
48. Menache, A.: *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann, San Francisco (1999)
49. Merwe, R.V.D., Doucet, A., Freitas, N.D., Wan, E.: The unscented particle filter. *Tech. Rep. F-INFENG/TR. 380*, CUED (2000)
50. Nakamura, Y., Hanafusa, H.: Inverse kinematic solutions with singularity robustness for robot manipulator control. *Trans. ASME Journal of Dynamic Systems, Measurement, and Control* **108**(3), 163–171 (1986)
51. O'Brien, J.F., Bodenheimer, R.E., Brostow, G.J., Hodgins, J.K.: Automatic joint parameter estimation from magnetic motion capture data. In: *Proceedings of Graphic Interface*, pp. 53–60 (2000)
52. Park, S.I., Hodgins, J.K.: Capturing and animating skin deformation in human motion. *ACM Transaction on Graphics (TOG)* **25**(3), 881–889 (2006)
53. Pechev, A.N.: Inverse kinematics without matrix inversion. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA, USA, pp. 2005–2012 (2008)
54. Phasespace inc: Optical motion capture systems. <http://www.phasespace.com>
55. Piazza, T., Lundström, J., Hugelstrand, A., Kunz, A., Fjeld, M.: Towards solving the missing marker problem in realtime motion capture. In: *Proceedings of the International Design Engineering Technical Conference* (2009)
56. Rhijn, A.V., Mulder, J.D.: Optical tracking and automatic model estimation of composite interaction devices. In: *IEEE Virtual Reality Conference*, pp. 135–142 (2006)
57. Ringer, M., Lasenby, J.: A procedure for automatically estimating model parameters in optical motion capture. In: *Proceedings of the British Machine Vision Conference*, pp. 747–756 (2002)
58. Rose, C., Cohen, M., Bodenheimer, B.: Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications* **18**(5), 32–40 (1998)
59. Silaghi, M.C., Plänklers, R., Boulic, R., Fua, P., Thalmann, D.: Local and global skeleton fitting techniques for optical motion capture. In: *Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, London, UK, pp. 26–40 (1998)
60. Singer, R.A.: Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems* **6**(1), 473–483 (1970)
61. Singer, R.A., Behnke, K.W.: Real-time tracking filter evaluation and selection for tactical applications. *IEEE Transactions on Aerospace and Electronic Systems* **7**(1), 100–110 (1971)
62. Sumner, R.W., Zwicker, M., Gotsman, C., Popović, J.: Mesh-based inverse kinematics. *ACM Transactions on Graphics (TOG)* **24**(3), 488–495 (2005)
63. Tak, S., Ko, H.S.: A physically-based motion retargeting filter. *ACM Transactions on Graphics (TOG)* **24**(1), 98–117 (2005)
64. Taylor, G.W., Hinton, G.E., Roweis, S.T.: Modeling human motion using binary latent variables. In: *Advances in Neural Information Processing Systems*, pp. 1345–1352. MIT Press, Cambridge (2007)
65. Wampler, C.W.: Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man and Cybernetics* **16**(1), 93–101 (1986)
66. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(2), 283–298 (2008)
67. Wang, L.C.T., Chen, C.C.: A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation* **7**(4), 489–499 (1991)
68. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., Colucci, D.: The HiBall tracker: high-performance wide-area tracking for virtual and augmented environments. In: *Virtual Reality Software and Technology, VRST*, pp. 1–10. ACM, New York (1999)

69. Wiley, D.J., Hahn, J.K.: Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications* **17**(6), 39–45 (1997)
70. Wolovich, W.A., Elliott, H.: A computational technique for inverse kinematics. *IEEE Conference on Decision and Control* **23**, 1359–1363 (1984)
71. Yu, Q., Li, Q., Deng, Z.: Online motion capture marker labeling for multiple interacting articulated targets. *Computer Graphics Forum (Proceedings of Eurographics)* **27**(7), 477–483 (2007)
72. Zordan, V.B., Van Der Horst, N.C.: Mapping optical motion capture data to skeletal motion using a physical model. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'03)*, pp. 245–250. Eurographics Association, San Diego (2003)



Andreas Aristidou is a post-doc researcher associated with the Department of Computer Science, University of Cyprus. Andreas has been awarded the ΔΙΑΚΤΩΡ fellowship, by the Cyprus Research Promotion Foundation, to establish research in motion analysis and classification; he is also collaborating with PhaseSpaceInc, a leading company that offers motion capture solutions for motion tracking and positioning. He had been a Cambridge European Trust fellow, at the signal processing group, University of

Cambridge, where he obtained his PhD. Andreas has a BSc in Informatics and Telecommunications from the National and Kapodistrian University of Athens (2005) and he is an honour graduate of Kings College London (2006), where he obtained his MSc degree in Mobile and Personal Communications. His research interests include human animation, motion capture, inverse kinematics, filtering and geometric algebra.



Joan Lasenby received her B.A & M.Math in Mathematics from the University of Cambridge (1983) and completed her Ph.D. in Radio Astronomy (at the Mullard Radio Astronomy Observatory, Cambridge) in 1987. She is currently a senior lecturer in the Signal Processing and Communications Group, Cambridge University Engineering Department. Her research interests include motion capture, computer vision, geometric algebra, and image processing.