

UNIVERSITY OF CAMBRIDGE

TECHNICAL REPORT

**Methods for real-time restoration and  
estimation in optical motion capture**

Andreas Aristidou, Joan Lasenby  
and Jonathan Cameron

CUED/F-INFENG/TR-619

January 14, 2009

© Cambridge University Engineering Department



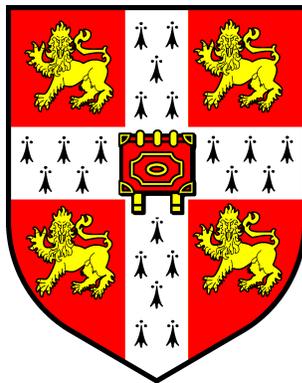
TECHNICAL REPORT

---

# METHODS FOR REAL-TIME RESTORATION AND ESTIMATION IN OPTICAL MOTION CAPTURE

**Andreas Aristidou, Joan Lasenby and Jonathan  
Cameron**

aa462@cam.ac.uk, jl221@cam.ac.uk, jic23@cam.ac.uk



Cambridge University  
Engineering Department  
Signal Processing and Communication Group

in Division F, October 2008

CUED/F-INFENG/TR-619

---

January 14, 2009

© Cambridge University Engineering Department



# Abstract

---

This report considers the problem of using marker locations from optical motion capture data to identify and parameterise the underlying human skeleton structure and motion over time. It is concerned with real-time algorithms suitable for use within a visual feedback system. The algorithms presented require 3 markers on each limb segment and can be implemented in a sequential fashion; hence the computational cost of updating the centres of rotation is independent of the number of data points previously available. However, a common problem in motion capture is marker occlusion. Most current methods are only useful for offline processing or become ineffective when a significant proportion of markers are missing for a long period of time. This report presents an integrated framework which predicts the occluded marker positions using a Kalman filter in combination with inferred information from neighbouring markers and thereby maintains a continuous data-flow. The results are accurate and reliable even in cases where all markers on a limb segment are occluded, or one or two markers are non-visible for a large sequence of frames. Moreover, it investigates several extensions of the proposed methodology for real-time applications using a sophisticated variable turn model, together with rotor prediction and unscented theory for filtering. It compares each individual method according to its accuracy, complexity and processing time and suggests the optimal solution for each case. Pre-defined models are not required and skeleton fitting to this complete data can then be updated in real-time.



# Contents

---

<b>Abstract</b>	<b>i</b>
<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Optical Motion Capture . . . . .	1
1.2. Literature Review and Motivation . . . . .	1
1.3. Outline of the Report . . . . .	2
<b>2. Obtaining 3D marker positions</b>	<b>5</b>
2.1. Apparatus . . . . .	6
2.2. Marker-to-limb association . . . . .	6
2.3. Rigid Body Dynamics . . . . .	7
<b>3. Real Time Joint Localisation</b>	<b>9</b>
3.1. Introduction . . . . .	9
3.2. Related Work . . . . .	9
3.3. Finding the Rotor between 2 sets of vectors using Unit Quaternions	11
3.4. Finding the Centre of Rotation (CoR) . . . . .	14
3.4.1. Cost function . . . . .	14
3.4.2. Minimising the Cost function . . . . .	15
<b>4. Estimating the occluded markers</b>	<b>21</b>
4.1. Related Work . . . . .	22
4.2. Kalman Filter . . . . .	23
4.3. Applying Constraints . . . . .	25
4.3.1. All markers are visible on a given limb . . . . .	25
4.3.2. One missing marker on a limb segment . . . . .	26

4.3.3.	Two missing markers on a limb segment . . . . .	27
4.3.4.	All markers on a limb segment are missing . . . . .	27
4.3.5.	Markers visible in only one camera . . . . .	29
<b>5.</b>	<b>Filtering extension using dynamic models and rotation prediction</b>	<b>31</b>
5.1.	Dynamic Model for Target Tracking . . . . .	31
5.1.1.	Introduction . . . . .	31
5.1.2.	Dynamic State-Space Model . . . . .	32
5.1.3.	Constant Turn Model . . . . .	32
5.1.4.	Variable Turn Model . . . . .	33
5.2.	Unscented Transform . . . . .	36
5.2.1.	Unscaled Unscented Transform . . . . .	37
5.2.2.	Scaled Unscented Transform . . . . .	39
5.2.3.	Unscented Kalman Filter . . . . .	41
5.3.	Rotation Prediction . . . . .	42
5.3.1.	Extrapolation . . . . .	43
5.3.2.	Prediction Filtering . . . . .	44
<b>6.</b>	<b>Results</b>	<b>47</b>
6.1.	The Experimental Environment . . . . .	47
6.2.	Results . . . . .	47
6.2.1.	One missing marker on a limb segment . . . . .	48
6.2.2.	Two missing markers on a limb segment . . . . .	48
6.2.3.	All markers on a limb segment are missing . . . . .	49
6.2.4.	Markers visible by one camera . . . . .	50
6.3.	The filtering extended methodologies . . . . .	53
6.4.	Discussion . . . . .	55
<b>7.</b>	<b>Conclusions and Future Work</b>	<b>61</b>
<b>A.</b>	<b>Alternative Cost Function</b>	<b>65</b>
	<b>List of Abbreviations</b>	<b>67</b>
	<b>Bibliography</b>	<b>69</b>

# List of Figures

---

2.1. Marker-to-limb association example with three markers on each limb. . . . .	6
2.2. Description of the rigid body. . . . .	7
3.1. The motion of two limbs with a time-difference of $m$ frames. . . .	13
3.2. A typical marker placement with relevant markers shown. . . . .	15
3.3. The rotors during frames. . . . .	20
3.4. The labeled markers positions and their CoRs. . . . .	20
4.1. The calculation of the observation vector in case where 2 markers are visible. . . . .	26
4.2. The calculation of the observation vector in case of only one visible marker. . . . .	27
4.3. The estimation procedure when all markers on a single limb segment are occluded. . . . .	28
4.4. The observation vector in the case of 2 visible markers and one marker visible only by one camera. . . . .	29
5.1. Target prediction with variable acceleration using the VTM model compared to the linear prediction. . . . .	35
5.2. The noisy marker's velocity and the same velocity after filtering using a median filter. . . . .	36
5.3. The acceleration using the noisy marker's velocity and the same acceleration using the filtered velocity. . . . .	37
5.4. The rotor trajectory over time. . . . .	44
5.5. The rotor trajectory and the rotation between two consecutive frames over time. . . . .	45
5.6. The marker's and Bivector components' trajectories over time. . . .	45
6.1. The Kalman filter error. . . . .	51
6.2. Examples of implementation on real data (lower Body). . . . .	57

6.3.	An example of the estimated and true position of markers and CoR location . . . . .	58
6.4.	The error between the predicted and the true positions of the (a) <i>Markers</i> and (b) <i>CoR</i> respectively during the occlusion period. . .	59
6.5.	The CDF of the estimation error. . . . .	60

# List of Tables

---

6.1.	Average results on real data with occlusions generated by deletions. Case of one missing marker on a limb segment. . . . .	49
6.2.	Average results on real data with occlusions generated by deletions. Case of two missing markers on a limb segment. . . . .	49
6.3.	Average results on real data with occlusions generated by deletions. Case where all markers on a limb segment are missing. . . . .	50
6.4.	Average results on real data with occlusions generated by deletions. Case of one missing marker on each limb segment. . . . .	52
6.5.	Average results on real data with occlusions generated by deletion. Case of one visible and two partially visible markers. . . . .	52
6.6.	Average results on real data with occlusions generated by deletion. Case of one entirely missing marker, one partially visible marker and one visible marker. . . . .	53
6.7.	The average error and its % change using a VTM instead of a CV model. . . . .	54
6.8.	The average error and its % reduction using a UKF (with VTM) instead of a KF (with CV). . . . .	54
6.9.	Average results after implementing the extension over the proposed methodology. . . . .	55



# 1

## Introduction

---

### 1.1. Optical Motion Capture

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. Such information can be used to better analyse techniques for sports training (e.g. posture, velocities, accelerations, angles, trajectories) [1]; a person's movements for medical reasons or sport performance [2]; to observe asymmetries, abnormalities in rehabilitation medicine (e.g. gait in stroke or prosthetic patients) [3, 4, 5]; in the generation of virtual characters for films or computer games [6]. There are two basic approaches used to capture such data, marked and markerless. Throughout this work, marked motion capture is used. The problem of establishing the motion of interest is simplified by attaching markers of some type to the subject being recorded. These markers can then be easily located in an image and their movement used to infer the movement of the person to whom they are attached.

### 1.2. Literature Review and Motivation

Tracking systems based on skeletal models require knowledge of the lengths of the skeleton's limbs, to which limb each marker is attached and the location of the markers relative to the underlying skeletal joints. So far, many techniques have been developed that estimate the model parameters by using three markers

on each limb. The approaches to *centre of rotation* (CoR) localisation can be separated into two distinct groups. The first group, often referred to as sphere fitting methods, assumes that all markers remain at a constant distance from the CoR, but makes no assumptions about relative marker positions [7, 8, 9, 10]. The second group is termed as transformational methods. This family of methods make the assumption that the markers on a given body segment are rigidly attached to each other [11, 12, 13, 14].

Many algorithms used to solve this problem first find a rigid body transform to move the problem into a frame of reference in which the limb segment on one side of the joint is stationary. A simpler one-sided problem is then solved. In this report, a technique based on the transformational method is further developed [14]. However, even with costly professional systems, there are instances where the system returns no data due to the occlusion of markers by limbs, bodies or other markers. Each marker must be visible to at least two cameras in each frame in order to unambiguously establish its position. Although many methods have been developed to handle the missing marker problem, most of them are not applicable in real-time and often require manual intervention.

In this report, a real-time approach for estimating the position of occluded markers using previous positions and information inferred from an approximate rigid body assumption has been proposed. 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. Thus, the neighbouring markers<sup>1</sup> provide us with useful information relevant to the current position of the non-visible marker. With a continuous stream of accurate 3D data, we can perform real-time centre of rotation (CoR) estimation, thereby producing skeletal information for use in visual performance feedback. Moreover, we present several filtering extensions using a variable turn model and the unscented transformation. A real-time rotor<sup>2</sup> prediction algorithm has been also examined using extrapolation and Kalman filtering. Finally, we compare the results of each method according to its accuracy, complexity and processing time discussing the optimum use of each method.

### 1.3. Outline of the Report

The body of this report may be broadly divided into 3 sections. Section 1 describes the mathematical and experimental framework. Chapter 2 describes the experimental environment including the cameras used for the experiments, the association between the markers and the limbs, and the clustering of marker data into groups corresponding to the limb segments to which they are attached.

---

<sup>1</sup>Neighbouring markers are considered as markers belonging to the same limb segment.

<sup>2</sup>Rotor are quantities that perform rotations in Geometric Algebra [15]

The second section is concerned with the problem of fitting skeletal models to marker-based optical motion capture data. Chapter 3 studies the problem of estimating the centres of rotation (joints) between every pair of limb segments and identifying the optimal skeleton in real-time. It presents an efficient and accurate method that uses unit quaternions for finding the rotors between 2 sets of vectors, and then, using these rotors, calculates the CoR.

Section 3 presents an integrated framework which predicts the occluded marker position and thereby maintains a continuous flow of data. Marker occlusion is a common phenomenon in motion capture systems due to camera system failure or marker occlusion by other limbs. Chapter 4 presents a real-time predicting approach using a Kalman filter in combination with inferred information from neighbouring markers in order to cope with cases where markers are missing. This approach takes advantage of the fact that markers located on the same limb of an articulated body have constant inter-marker distance. Chapter 5 describes filtering extensions of the previously proposed model using a variable turn model, as the transition function of the filter, and the unscented transformation within a Kalman filter for marker predictions. Finally we consider rotor prediction algorithms using extrapolation and filtering approaches. Chapter 6 presents, compares and discusses the results of the methods proposed in Chapters 4 and 5.

Finally, Chapter 7 presents conclusions, the optimal solution for each case and suggestions for future work.



# 2

## Obtaining 3D marker positions

---

Motion capture hardware, such as that provided by Phasespace [16], CodaMotion [17] and Vicon [18], is under constant development, providing real-time acquisition of labelled 3D marker data. These data can be used for reconstruction of the human skeleton allowing accurate real-time feedback via tracking and modelling of human motion.

Optical motion capture has many applications. It can be used to better analyse techniques for sports training [1]; to observe asymmetries and abnormalities in rehabilitation medicine (clinical analysis) [3, 4, 5]; for biomechanics (prosthetics, ergonomics); to study the person's movements for medical reasons or sport performance [2]; for gait labs; or for visualisation of virtual characters for films and computer games [6]. Throughout this work, marked motion capture system will be used in order to provide information related to the human skeleton and for localisation of the centres of rotation.

This chapter presents the apparatus of the experimental environment. Thereafter, it provides more details related to marker-to-limb association, which is very important for the accurate performance of the system. Finally, it explains the rigid body dynamics of a body position in terms of rotation and translation from a 'reference' body position, that will be regularly used in later chapters.

## 2.1. Apparatus

In order to overcome the tracking problem, we used a 16 camera and 480 *frames per second* motion capture system using modulated LEDs, provided by Phasespace [16]. These cameras contain a pair of linear scanner arrays operating at high frequency each of which can capture the position of any number of bright spots of light as generated by the LEDs. The markers consist of a control circuit and bright red LEDs which are wired to a synchronisation box.

For reconstruction it is necessary for each marker to be visible by at least two cameras in each frame. The system offers a fast rate of capture (480Hz) and allows the individual markers to be identified by combining the information from several frames and hence identifying the marker from its unique modulation. The markers are placed at strategic points on the articulated body so that these points can be easily and accurately located by the cameras. It is desirable to allocate the markers such that there is the same number of markers on each limb, if possible, in order to obtain the same quality of data for all parts of the body we are interested in. The subject moves in a specified space that can be tracked by the cameras and the markers attached to its body are tracked over time and used to reconstruct the three-dimensional pose of the subject at each instant of time.

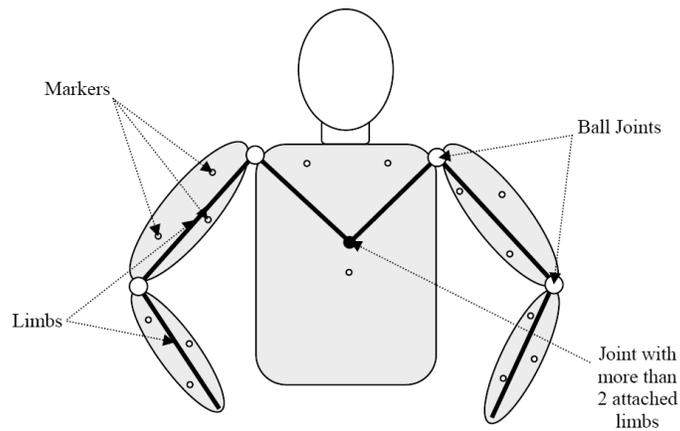


Figure 2.1.: Marker-to-limb association example with three markers on each limb. This example shows the upper body with the marker-to-limb association.

## 2.2. Marker-to-limb association

Since the three-dimensional trajectory of each marker is available we are able to determine which markers are attached to which limb on the body. This is done in two steps:

1. Firstly, markers attached to the same limb are grouped. This is done by finding which markers maintain the same distance from each other throughout the motion. Since the data is noisy we expect that the distance between the markers does not remain constant. Therefore, the variances of the distances between markers are calculated and the markers are clustered as belonging to the same limb if the variance of the distance between them is less than a certain threshold.
2. Afterwards, association of the groups of markers to specific limbs of the body is necessary. This is done either manually or automatically by first calculating the distance between the centroid of each group of markers in a given time frame and then, according to the model used, the limbs are identified. It is assumed that the skeleton consists of rigid limbs connected with ball joints.

The approach described above was implemented in MATLAB [19] with data taken using the Phasespace equipment to find the marker-to-limb association. Despite the occlusions and the noisy data, the correct associations were obtained. An example of associated markers is presented in figure 2.1.

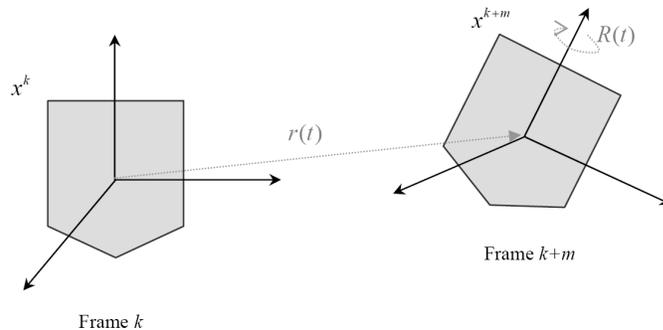


Figure 2.2.: Description of the rigid body. The vector  $r(t)$  specifies the position of the centre of mass, relative to the origin. The rotor  $R(t)$  defines the orientation of the body, relative to a fixed copy placed at the origin. The  $x^k$  is a vector in a reference body (in this example in frame  $k$ ), and  $x^{k+m}$  is a vector in space of the equivalent point of the moving body (in this example after  $m$  frames).

## 2.3. Rigid Body Dynamics

A rigid body can be viewed as a system of particles moving subject to the constraint that all inter-particle distances are fixed. The final body position can be expressed in terms of a rotation and translation from a fixed “reference” body on to the body in space (figure 2.2). We let  $r(t)$  denotes the position of the centre

of rotation and  $x^{k+m}$  denote the position in space of a point in the body. These are related by:

$$x^{k+m} = Rx^k\tilde{R} + r(t) \quad (2.1)$$

where  $x^k$  is a fixed constant vector in the reference copy of the body. In this manner we have placed all of the rotational motion in the time-dependent rotor  $R(t)$ . For more information about rotors and geometric algebra (GA) refer to [15].

# 3

## Real Time Joint Localisation

---

### 3.1. Introduction

The problem of automatic skeleton generation can be separated into three stages. First is the marker clustering, then the problem of finding the joint location and finally, the last stage is the identification of the full skeleton. This chapter will be focused on the second stage, that is the joint localisation, which is also termed Centre of Rotation (CoR) localisation.

### 3.2. Related Work

Several papers have focused on methods for localisation of the centre of rotation. These methods can be separated into two major groups. In the first group, also known as *sphere fitting* methods, it is assumed that all the markers remain a constant distance from the CoR, but that we make no assumptions about the positions of the markers belonging to the same limb. The second group of methods, known as *transformational* methods, assumes all positions of the markers on a body segment are related to each other in some way.

#### **Sphere fitting**

Sphere fitting methods 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. A first step for solving this problem is to find a rigid body transformation moving the problem into a reference frame (e.g. *frame 1*) so that the limb segment on one side of the joint is viewed as stationary and then solve a simpler one-sided problem. However, this approach returns an approximation of the CoR because the transform is calculated from the marker position measurements. Dorst, in [20], presents a first order approximation which also illustrates the dependence of the attitude estimation error on the distribution of the point cloud (in this case the marker locations on the limb segments).

In [7] the Levenberg-Marquardt method was implemented in order to optimise the location of the centre of rotation and the radii of the marker spheres. A cost function  $S = (|x_m^k - C| - r_m)^2$  was introduced, where  $x_m^k$  is the marker position of the marker  $m$  at frame  $k$ ,  $C$  is the centre of the sphere and  $r_m$  is the radius of the sphere associated with the marker  $m$ . The overall cost is the sum of the individual costs over all markers and all frames. This algorithm requires a series of weights to be set, such as perceived accuracies of markers and a spatial reweighting of data points via a voxel grid. This method also has disadvantages; the inaccuracy of the heuristics used to set the weight parameters and the non-linear nature of the solver, making it susceptible to problems with local minima.

Halvorsen et al., in [8], describe a closed form solution using the geometric properties of the sphere. They used the fact that the perpendicular bisectors of chords of a sphere intersect at the sphere origin, and every pair of frames provide an approximate perpendicular bisector for each marker. However, this method is dependent on which data points are used to form the chords, affecting the accuracy and effectiveness of the algorithm.

In [21], Gamage et al. also introduced a closed form solution, using a cost function of the squared differences in the squared distance from the CoR  $C$ , to a marker  $m$ , at position  $x_m^k$  in frame  $k$ , and the radius of the sphere associated with the marker  $r_m$ . That is

$$((C - x_m^k)^2 - (r_m)^2)^2 \quad (3.1)$$

Cereatti et al, [9], shows that, the cost function used in [21] corresponds to the minimised cost function used in [8], when all possible chords are used.

An alternative approach provided by Halvorsen, [10], gives a Bayesian analysis of the algorithm of [21], providing a first order approximation of the effect of isotropic Gaussian noise upon the algorithm. An extension to [10] can be achieved by assuming that the measured points are the result of Gaussian noise only in

the axial direction<sup>1</sup>. Hence, the cost function of [21] can be written as:

$$\begin{aligned}
 S &= \sum_k \sum_m ((e_m^k + r_m)^2 - (r_m)^2) \\
 &= \sum_k \sum_m ((e_m^k)^2 + 2r_m e_m^k)
 \end{aligned} \tag{3.2}$$

where the error  $e_m^k$  is equal to  $e_m^k = \sqrt{(C - x_m^k)^2} - r_m$ . Therefore, the cost value is dependent on the radii  $r_m$  and the error  $e_m^k$  so the algorithm will tend to underestimate the radii due to the second term,  $2r_m e_m^k$ , being linearly dependent on the radii.

The result of the algorithm proposed in [21] was recently used in Cerveri et al. ([22], [23]) who attempted to find an optimal fit for a hand skeleton.

### Transformational

The transformational method assumes that all positions of the markers are rigidly attached to limb segments. Such an approach was implemented in [11]. A technique for using magnetic motion capture data to determine the joint parameters of an articulated hierarchy was described in [12]. This technique makes it possible to determine limb lengths, joint locations, and sensor placement for a human subject without external measurements, just from the motion data acquired during the capture session. The parameters are computed by performing a linear least squares fit of a rotary joint model to the input data. In many recent papers, such as [13], the same techniques are applied but the orientation is obtained from sets of optical markers.

In [14] a sequential algorithm was presented to locate the rotation centres of a human skeleton from marker data assuming that all markers on a body segment are attached to a rigid body. This method does not suffer from optimisation steps with computational requirements that grow with the amount of data supplied ([21], [24]) and no user feedback to set marker weights, as in [7], are needed. The method is closed form, thus enabling real-time implementation.

## 3.3. Finding the Rotor between 2 sets of vectors using Unit Quaternions

Assuming that labelled data are available (through the Phasespace system which is used in our experiments), and the tracking problem is therefore solved, the next step is to divide the markers into sets of markers attached to each limb segment. This can be achieved using a cost function along with a number of

---

<sup>1</sup>The axial direction is that parallel to a line from the centre of the sphere to the true marker location

different clustering techniques. The next stage is the localisation of the joints in each frame. Before the localisation of the joints, we have the problem of how to extract the rotation of two limbs between frames. We therefore seek an optimal estimation of the rotor expressing the rotation between the marker locations of a given limb segment ('referring' frame) and the new rotated locations on future frames. The algorithm used for finding estimates of rotors within this report, is that proposed by Horn [25].

If we take a set of labelled points  $\{x_i\}$  and the same set of points after an unknown rotation  $R(t)$ ,  $\{w_i\}$ , then the problem of finding the unknown rotor (unit quaternion)  $R$  can be formulated as,

$$R = \arg \max \sum_{i=1}^n \left( R x_i \tilde{R} \right) \cdot w_i \quad (3.3)$$

In order to avoid introducing quaternion terminology and methods, we can express the solution in terms of Geometric Algebra (for more details about GA see [15]). To clarify the following equations, the marker index  $i$  will temporally be suppressed. Let:

$$R = a + B_1 e_{23} + B_2 e_{13} + B_3 e_{12} \quad (3.4)$$

$$x_i = x_1 e_1 + x_2 e_2 + x_3 e_3 \quad w_i = w_1 e_1 + w_2 e_2 + w_3 e_3 \quad (3.5)$$

where  $a$  is a scalar,  $e_1, e_2, e_3$  are orthogonal vectors in the 3-D space, and  $e_{23}, e_{31}$  and  $e_{12}$  are the unit bivectors as reported in [26]. Each of these encodes a distinct plane, and there are 3 of them to match the 3 independent orthogonal planes in the 3-D space.

Then:

$$\begin{aligned} (R x_i \tilde{R}) \cdot w_i &= a^2 (x \cdot w) + 2aB_1(x_3w_2 - x_2w_3) + 2aB_2(x_3w_1 - x_1w_3) \\ &\quad + 2aB_3(2x_2w_1 - 2x_1w_2) + B_1^2(x_1w_1 - x_2w_2 - x_3w_3) \\ &\quad + 2B_1B_2(-x_2w_1 - x_1w_2) + 2B_1B_3(x_3w_1 + x_1w_3) \\ &\quad + B_2^2(-x_1w_1 + x_2w_2 - x_3w_3) + 2B_2B_3(-x_3w_2 + x_2w_3) \\ &\quad + B_3^2(-x_1w_1 - x_2w_2 + x_3w_3) \end{aligned}$$

This can be expressed as follows:

$$R = \arg \max \sum_{i=1}^n (R_v^T N R_v) \quad (3.6)$$

where:

$$R_v = \begin{bmatrix} a \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

$$N = \begin{bmatrix} x_1 w_1 + x_2 w_2 + x_3 w_3 & x_3 w_2 - x_2 w_3 & x_3 w_1 - x_1 w_3 & x_2 w_1 - x_1 w_2 \\ x_3 w_2 - x_2 w_3 & x_1 w_1 - x_2 w_2 - x_3 w_3 & -(x_2 w_1 + x_1 w_2) & x_3 w_1 + x_1 w_3 \\ x_3 w_1 - x_1 w_3 & -(x_2 w_1 + x_1 w_2) & -x_1 w_1 + x_2 w_2 - x_3 w_3 & x_3 w_2 - x_2 w_3 \\ x_2 w_1 - x_1 w_2 & x_3 w_1 + x_1 w_3 & x_3 w_2 + x_2 w_3 & -x_1 w_1 - x_2 w_2 + x_3 w_3 \end{bmatrix}$$

In terms of limb motion we note that  $x_1, x_2, x_3$  and  $w_1, w_2, w_3$  correspond to values given below, with  $x_i^k, y_i^k, c_x^k$  and  $c_y^k$  defined in figure 3.1.

$$\begin{aligned} x_1 &= x_1^k - c_x^k & w_1 &= x_1^{k+m} - c_x^{k+m} \\ x_2 &= x_2^k - c_x^k & w_2 &= x_2^{k+m} - c_x^{k+m} \\ x_3 &= x_3^k - c_x^k & w_3 &= x_3^{k+m} - c_x^{k+m} \end{aligned} \quad (3.7)$$

$$c_x^k = \frac{x_1^k + x_2^k + x_3^k}{3} \quad c_x^{k+m} = \frac{x_1^{k+m} + x_2^{k+m} + x_3^{k+m}}{3} \quad (3.8)$$

As a valid rotor must always obey the property  $R\tilde{R} = 1$  it must be the case that  $R_v^T R_v = 1$ . If this condition is relaxed and normalisation is introduced into the above matrix expression we have:

$$R = \arg \max \left( \frac{R_v^T \sum_{i=1}^n (N) R_v}{R_v^T R_v} \right) \quad (3.9)$$

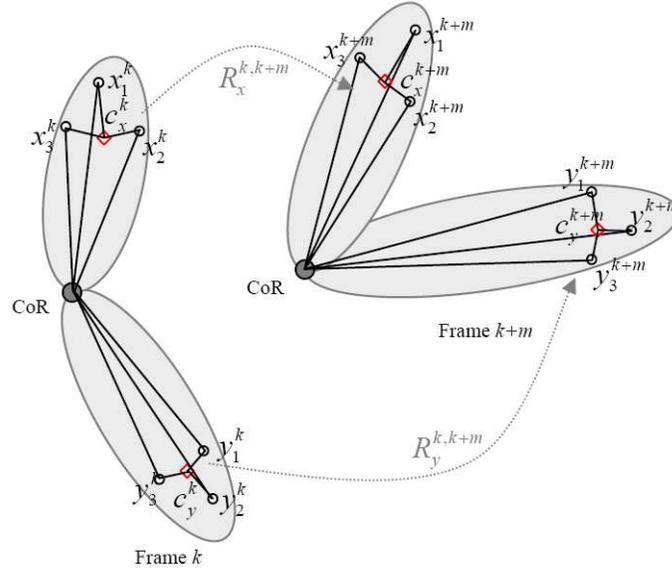


Figure 3.1.: The motion of two limbs with a time-difference of  $m$  frames.

This expression is now in the form of the Rayleigh Quotient;  $\sum_{i=1}^n (N)$  is clearly Hermitian and the value of  $R$  at the maximum value of the expression to be

maximised. Thus, provides a least squares estimate of the required rotor, where the eigenvector associated with the greatest eigenvalue maximises the matrix product. Therefore the rotor  $R_v$  which corresponds to the rotation is equal to that eigenvector.

### 3.4. Finding the Centre of Rotation (CoR)

Locating the CoRs is important in both computer graphics and rehabilitation medicine. It is a crucial step in acquiring a skeleton from raw motion capture data. In this work we used an algorithm based on the method in [14], which considers a formulation of the problem that takes full advantage of the simplification that all markers on a body segment are attached to a rigid body. The assumptions required for implementing this procedure are:

- the joint sought is between two rigid bodies, and
- each rigid body has an non-degenerate set of markers.

We further assume that each limb has at least three markers, which is the minimum number of markers for precise calculation of the rotation.

#### 3.4.1. Cost function

The method used to estimate the centres of rotation is a variation of the least squares solution proposed in [27]. The main principles are illustrated below.

Suppose we place markers on two joint limb segments ( $x$  and  $y$ ) of a body. Let that body perform some movements so that the markers can be tracked easily over time by the cameras. Let the centre of rotation for a pair of joint limb segments in frame  $k$  be  $C^k$  and the vectors from the centre of rotation to markers in the reference frame, be denoted by  $a_x^i$  for limb  $x$  and  $a_y^j$  for limb  $y$  where  $i$  and  $j$  are labels for which marker we are referring to.

Given the set of 3D coordinates of each marker over time and given that we already know which limb each marker is attached to, we try to find the CoR of the joint. We initially consider each joint and the pair of limbs rotating about it independently. The position of the markers in frame  $k$  is given by:

$$x_i^k = C^k + R_x^k a_x^i \tilde{R}_x^k \quad y_j^k = C^k + R_y^k a_y^j \tilde{R}_y^k \quad (3.10)$$

Let  $b_{ij}^k$  be the vector from  $y_j^k$  to  $x_i^k$ , that is:

$$b_{ij}^k = x_i^k - y_j^k = R_x^k a_x^i \tilde{R}_x^k - R_y^k a_y^j \tilde{R}_y^k \quad (3.11)$$

Figure 3.2 presents a typical marker placement with the  $b_{ij}^k$ ,  $a_x^i$  and  $a_y^j$  as shown.

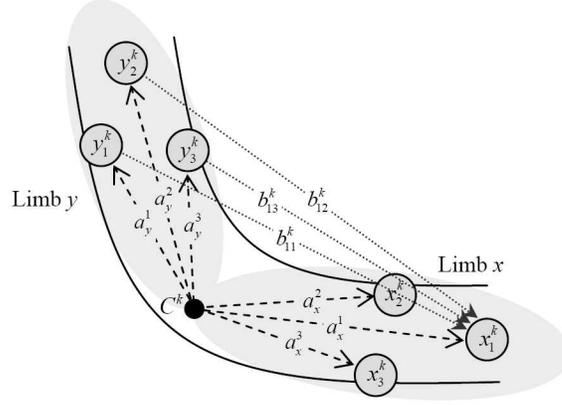


Figure 3.2.: A typical marker placement with relevant markers shown.

Now a cost function  $S$  can be constructed that has a global minimum at the correct values of  $a_x^i$  and  $a_y^j$  if the data is noise free. As we shall see, this minimum is a good estimate in the presence of moderate noise on the readings of marker positions.

$$S = \sum_{k=1}^m \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \left[ b_{ij}^k - \left( R_x^k a_x^i \tilde{R}_x^k - R_y^k a_y^j \tilde{R}_y^k \right) \right]^2 \quad (3.12)$$

where  $n_x$  is the number of markers on limb  $x$ ,  $n_y$  is the number of markers on limb  $y$  and  $m$  is the number of frames used for the calculations.

### 3.4.2. Minimising the Cost function

The cost function can be expanded to give:

$$S = \sum_{k=1}^m \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \left[ (b_{ij}^k)^2 - 2b_{ij}^k \cdot \left( R_x^k a_x^i \tilde{R}_x^k - R_y^k a_y^j \tilde{R}_y^k \right) + (a_x^i)^2 + (a_y^j)^2 - 2R_x^k a_x^i \tilde{R}_x^k \cdot R_y^k a_y^j \tilde{R}_y^k \right]$$

Note that under the assumption that there are at least 3 markers on each limb, each of the  $R_x^k$  and  $R_y^k$  can be reconstructed in a least-squared sense. Now differentiating w.r.t. the  $a_x^i$  and  $a_y^j$  vectors using the standard results  $a \cdot Rb\tilde{R} = \tilde{R}aR \cdot b$  and  $\partial_a (a \cdot b) = b$ .

$$\partial_{a_x^i} S = 2 \sum_{k=1}^m \sum_{j=1}^{n_y} \left[ -\tilde{R}_x^k b_{ij}^k R_x^k + a_x^i - \tilde{R}_x^k R_y^k a_y^j \tilde{R}_y^k R_x^k \right] \quad (3.13)$$

$$\partial_{a_y^j} S = 2 \sum_{k=1}^m \sum_{i=1}^{n_x} \left[ -\tilde{R}_y^k b_{ij}^k R_y^k + a_y^j - \tilde{R}_y^k R_x^k a_x^i \tilde{R}_x^k R_y^k \right] \quad (3.14)$$

An alternative derivation of the above is presented in Appendix A.

In order to get the minimum of the cost function, we require the partial derivatives to be equal to 0. Thus, from equation 3.13 we obtain:

$$\begin{aligned}
\sum_{k=1}^m \sum_{j=1}^{n_y} a_x^i &= \sum_{k=1}^m \sum_{j=1}^{n_y} \left\{ \tilde{R}_x^k b_{ij}^k R_x^k + \tilde{R}_x^k R_y^k a_y^j \tilde{R}_y^k R_x^k \right\} \\
mn_y a_x^i &= \sum_{k=1}^m \sum_{j=1}^{n_y} \left\{ \tilde{R}_x^k b_{ij}^k R_x^k + \tilde{R}_x^k R_y^k a_y^j \tilde{R}_y^k R_x^k \right\} \\
a_x^i &= \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k b_i^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k
\end{aligned} \tag{3.15}$$

$$\begin{aligned}
\bar{a}_x &= \frac{1}{n_x} \sum_{i=1}^{n_x} \left[ \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k b_i^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k \right] \\
\bar{a}_x &= \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k
\end{aligned} \tag{3.16}$$

where

$$\bar{a}_x = \frac{1}{n_x} \sum_{i=1}^{n_x} a_x^i \qquad \bar{a}_y = \frac{1}{n_y} \sum_{j=1}^{n_y} a_y^j \tag{3.17}$$

$$b_i^k = \frac{1}{n_y} \sum_{j=1}^{n_y} b_{ij}^k \qquad b_j^k = \frac{1}{n_x} \sum_{i=1}^{n_x} b_{ij}^k \tag{3.18}$$

and

$$\bar{b}^k = \frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} b_{ij}^k \tag{3.19}$$

Similarly, from equation 3.14:

$$\begin{aligned}
\sum_{k=1}^m \sum_{i=1}^{n_x} a_y^j &= \sum_{k=1}^m \sum_{i=1}^{n_x} \left\{ -\tilde{R}_y^k b_{ij}^k R_y^k + \tilde{R}_y^k R_x^k a_x^i \tilde{R}_x^k R_y^k \right\} \\
mn_x a_y^j &= \sum_{k=1}^m \sum_{i=1}^{n_x} \left\{ -\tilde{R}_y^k b_{ij}^k R_y^k + \tilde{R}_y^k R_x^k a_x^i \tilde{R}_x^k R_y^k \right\} \\
a_y^j &= \frac{1}{m} \sum_{k=1}^m -\tilde{R}_y^k b_j^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k
\end{aligned} \tag{3.20}$$

$$\begin{aligned}
\bar{a}_y &= \frac{1}{n_y} \sum_{i=1}^{n_y} \left[ \frac{1}{m} \sum_{k=1}^m -\tilde{R}_y^k b_j^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k \right] \\
\bar{a}_y &= \frac{1}{m} \sum_{k=1}^m -\tilde{R}_y^k \bar{b}^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k
\end{aligned} \tag{3.21}$$

Substituting equation 3.21 into equation 3.16 we obtain:

$$A_1 \bar{a}_x = d_1 \quad (3.22)$$

where:

$$\begin{aligned} A_1 &= \left[ I - \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_x^k R_y^k \left( \sum_{k'=1}^m \tilde{R}_y^{k'} R_x^{k'} \tilde{R}_x^{k'} R_y^{k'} \right) \tilde{R}_y^k R_x^k \right\} \right] \\ d_1 &= \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k - \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_x^k R_y^k \left( \sum_{k'=1}^m \tilde{R}_y^{k'} \bar{b}^{k'} R_y^{k'} \right) \tilde{R}_y^k R_x^k \right\} \end{aligned} \quad (3.23)$$

Similarly, substituting the equation 3.16 into equation 3.20 we obtain:

$$A_2 \bar{a}_y = d_2 \quad (3.24)$$

where:

$$\begin{aligned} A_2 &= \left[ I - \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_y^k R_x^k \left( \sum_{k'=1}^m \tilde{R}_x^{k'} R_y^{k'} \tilde{R}_y^{k'} R_x^{k'} \right) \tilde{R}_x^k R_y^k \right\} \right] \\ d_2 &= -\frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k \bar{b}^k R_y^k + \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_y^k R_x^k \left( \sum_{k'=1}^m \tilde{R}_x^{k'} \bar{b}^{k'} R_x^{k'} \right) \tilde{R}_x^k R_y^k \right\} \end{aligned} \quad (3.25)$$

Then,  $\bar{a}_x$  and  $\bar{a}_y$  are found using the pseudo-inverse, which provides us with the Least Square solutions:

$$\bar{a}_x = (A_1^T A_1)^{-1} A_1^T d_1 \quad \bar{a}_y = (A_2^T A_2)^{-1} A_2^T d_2 \quad (3.26)$$

Since  $R_x^k a_x^i \tilde{R}_x^k = x_i^k - C^k$  (equation 3.10), then:

$$\begin{aligned} R_x^k \bar{a}_x \tilde{R}_x^k &= \frac{1}{n_x} \sum_{i=1}^{n_x} x_i^k - C^k \\ \Rightarrow C^k &= \frac{1}{n_x} \sum_{i=1}^{n_x} x_i^k - R_x^k a_x^i \tilde{R}_x^k \end{aligned} \quad (3.27)$$

Similarly,

$$\begin{aligned} R_y^k \bar{a}_y \tilde{R}_y^k &= \frac{1}{n_y} \sum_{j=1}^{n_y} y_j^k - C^k \\ \Rightarrow C^k &= \frac{1}{n_y} \sum_{j=1}^{n_y} y_j^k - R_y^k a_y^j \tilde{R}_y^k \end{aligned} \quad (3.28)$$

Hence,

$$C^k = \frac{1}{2n_x} \sum_{i=1}^{n_x} x_i^k + \frac{1}{2n_y} \sum_{j=1}^{n_y} y_j^k - \frac{1}{2} \left( R_x^k \bar{a}_x \tilde{R}_x^k + R_y^k \bar{a}_y \tilde{R}_y^k \right) \quad (3.29)$$

The above solution is a matrix-based approach. We observed that equations 3.16 and 3.21 are a set of simultaneous equations and so can be directly solved via the inversion of a  $3n_x + 3n_y$  matrix. The exact form of the matrix equation solved is given below:

$$\begin{aligned} m a_x^i &= \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k + \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k \\ m a_x^i - \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k &= \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k \end{aligned} \quad (3.30)$$

Similarly,

$$\begin{aligned} m a_y^j &= \sum_{k=1}^m \tilde{R}_y^k \bar{b}^k R_y^k + \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k \\ m a_y^j - \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k &= \sum_{k=1}^m \tilde{R}_y^k \bar{b}^k R_y^k \end{aligned} \quad (3.31)$$

Thus, we have  $n_x$  3D linear vector equations in  $a_x^i$ , so the problem can be reformulated as:

$$M a = b \quad (3.32)$$

where

$$M = \begin{bmatrix} mI_3 & M_x \\ M_y & mI_3 \end{bmatrix} \quad (3.33)$$

and

$$\begin{aligned} M_x &= \begin{bmatrix} -\sum_k \tilde{R}_x^k R_y^k e_1 \tilde{R}_y^k R_x^k & -\sum_k \tilde{R}_x^k R_y^k e_2 \tilde{R}_y^k R_x^k & -\sum_k \tilde{R}_x^k R_y^k e_3 \tilde{R}_y^k R_x^k \end{bmatrix} \\ M_y &= \begin{bmatrix} -\sum_k \tilde{R}_y^k R_x^k e_1 \tilde{R}_x^k R_y^k & -\sum_k \tilde{R}_y^k R_x^k e_2 \tilde{R}_x^k R_y^k & -\sum_k \tilde{R}_y^k R_x^k e_3 \tilde{R}_x^k R_y^k \end{bmatrix} \\ a &= \begin{bmatrix} \bar{a}_x \\ \bar{a}_y \end{bmatrix}, \quad b = \begin{bmatrix} \sum_k \tilde{R}_x^k \bar{b}^k R_x^k \\ -\sum_k \tilde{R}_y^k \bar{b}^k R_y^k \end{bmatrix} \text{ and } e_1, e_2, e_3 \text{ are the three basis vectors.} \end{aligned}$$

A clear saving in computational expense can be made by noticing that  $M_x = M_y^T$ .

If a simple average is to be used of the estimate of the joint, the size of the matrix requiring inversion can be reduced to a constant  $6 \times 6$  by applying the above algorithm to find the pair of vectors from the centre of mass of the markers on each limb segment to the CoR. The form of the matrix allows for an efficient block based inverse to be applied to further reduce the required computation. The block based inverse of equation 3.33 can be achieved using the following transformation:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix} \quad (3.34)$$

where  $S_A = (D - CA^{-1}B)$  is the Schur complement of  $A$ . Clearly  $A^{-1}$  is trivial, so the only computationally expensive matrix inverse needed is  $S_A^{-1}$  that is simply the inverse of a  $3 \times 3$  matrix. Thus we exchange a single  $6 \times 6$  matrix inverse for a  $3 \times 3$  one and 5 general matrix multiplies, a single full matrix addition and 4 matrix scalings. This algorithm can be implemented in a sequential fashion and as such the computational cost of updating the centres of rotation is independent of the number of data points previously available. Hence, this method is closed form, and therefore more practical and computationally less costly, enabling real-time implementation.

As can be seen from equations 3.16 and 3.21, all of the terms, in the linear system to be solved, can be sequentially updated as more data becomes available and the only step which much be repeated at each stage is the actual matrix inversion.

In a real-time system the choice of base frame, the data frame relative to which all rotors are computed, is restricted to those that are available. As it is desirable to compute an estimate of the CoRs as quickly as possible, the obvious choice would be to simply take the first frame in which all the markers on the relevant pair of limb segments are present. However, to avoid any unnecessary re-computation of rotors and to maintain the sequential nature of the algorithm, all rotors are computed as the compound of a rotor taking a given frame back to the first frame in which all the markers of the single relevant cluster are present and a rotor to the first frame in which both sides are present. Figure 3.3 shows how rotors change during frames.

When the centres of rotations are found, the limb lengths can be estimated by averaging the distance between the trajectories of each joint, as reported in [28]. Figure 3.4(a) shows the labeled marker positions of a lower body as returned by the Phasespace system. Figure 3.4(b) shows the same markers with their CoRs and the limb segments the markers are attached to, implemented in real-time using the above methodology. Having established the necessary parameters, a skeletal model can be fitted to the data. However, due to occlusions, there are instances where not all markers are available. Using the above procedure we can estimate the position of the centre of rotation  $\hat{C}^k$  using the  $R_w^k$  and the last available value of  $\bar{a}_w$ , where  $w = \{x, y\}$  (assuming that its value does not change

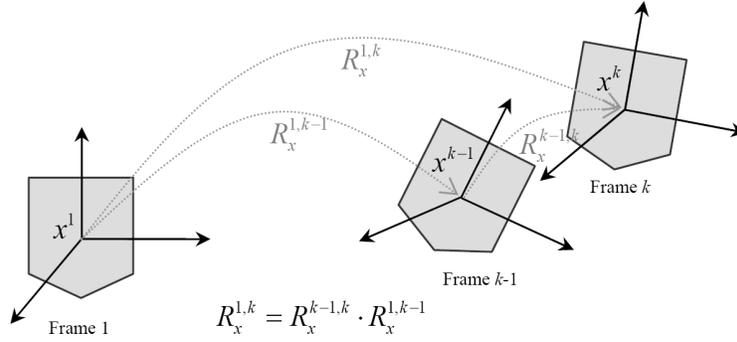


Figure 3.3.: The rotors during frames.

over time). Hence, using eq. 3.10 the CoR can be estimated using information from the limb segment which has all its markers visible. Thus,

$$\hat{C}^k = w_i^k - R_w^k a_w^i \tilde{R}_w^k \quad (3.35)$$

This approach gives accurate CoR estimates only in cases where one limb segment has all its markers available, otherwise it fails to return the CoR. Therefore, a method of handling cases with marker occlusions is needed.

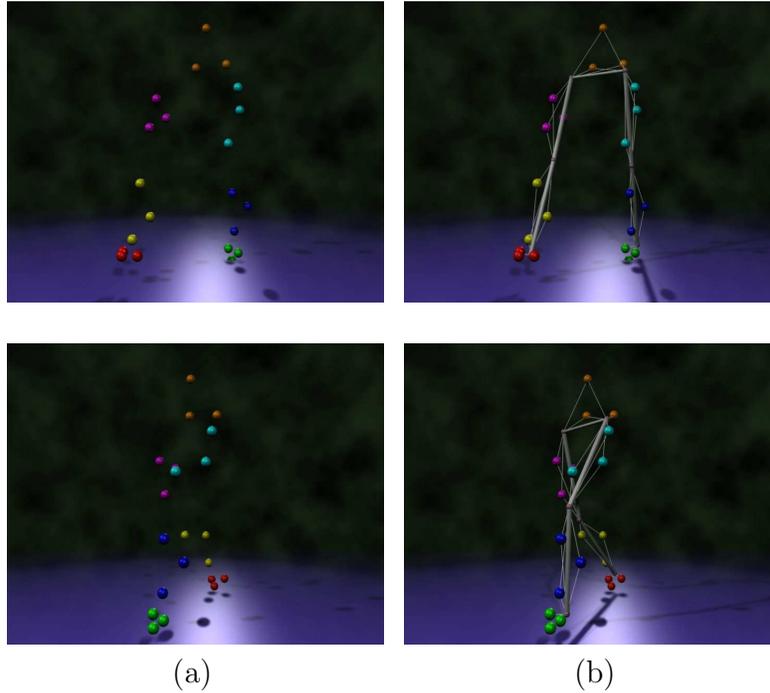


Figure 3.4.: (a) The labeled markers positions, as returned by the Phasespace system, (b) the marker positions with their CoRs and limb segments.

# 4

## Estimating the occluded markers

---

The Phasespace [16] system provides 16 cameras and modulated LEDs. Each marker must be visible by at least two cameras in each frame in order to return its 3D position. Hence, the markers are placed at strategic points on the articulated body so that these points can be easily and accurately located by the cameras. However, a common phenomenon encountered in motion capture is that markers return ‘NULL’ values due to occlusions or ambiguities in the system, and there are thus frames with partial data. A major cause of missing markers is that a marker is occluded by limbs, bodies, props or other markers. As has been seen, for the accurate calculation of the centre of rotation between two limbs, three markers on each limb are required. As missing data can cause failure in the CoR estimation, it is essential to find a way of ‘recapturing’ the missing data.

This section describes an integrated framework which produces real-time prediction of the missing markers in order to drive real-time centre of rotation estimation. The proposed approach uses a Kalman filter in combination with inferred information from neighbouring markers, taking advantage of the fact that markers on a given limb segment have constant inter-marker distance.

Hereafter, in this report, vectors will be designated in bold font to distinguish them from other symbols and avoid any confusion.

## 4.1. Related Work

A number of methods that can deal with this problem have been proposed, but these do not generally run in real time and often require manual intervention. Also, many of these methods behave suboptimally with diverse motions, a high percentage of missing markers and/or external occlusions. A popular method is the interpolation of the missing data using linear or non-linear approaches [29, 30, 31, 32, 33]; this can produce very accurate results, but is useful only in post-processing. Another drawback of the interpolation methods is that they can effectively estimate the missing markers if they are missing for only a short period of time, typically less than 0.5 *seconds*. Some MoCap systems also provide missing markers recovery solutions using interpolation techniques in combination with kinematic information, but they do not reliably work in real-time.

Another commonly used method is to ignore the frames with ‘NULL’ data and use only the frames which have useful information. This approach is very simple but there exist cases where the omission of specific data could lead to the loss of useful information. Also, long running occlusions leading to a large sequence of missing data or when a significant portion of markers are missing for extended period of time, can cause complete failure of the system.

Rhijn and Mulder, [34], proposed a novel model-based optical tracking and model estimation system for composite interaction devices. The proposed system automatically constructs the geometric skeleton structure, degrees of freedom (DOF) relations, and DOF constraints between segments, and thus pre-defined models are not required. The system supports segments with only a single marker, so that interaction devices can be small with a low number of markers. However, it is an off-line procedure and cannot be used in real-time applications.

Dorfmueller in [35] used an extended Kalman filter (EKF) to predict the missing markers using previously available marker information while Welch et al. in [36] used an EKF to resolve occlusions based on the skeletal model of the tracked person. Again, these methods require manual intervention or become ineffective in cases where markers are missing for an extended period of time.

Herda et al., in [37] and [38], used a post-processing approach to increase the robustness of a motion capture system by using a sophisticated human model. They can predict the 3D location and visibility of the markers increasing the robustness of the marker tracking and reducing the need for human intervention during the reconstruction process. The neighboring markers that share kinematic relations with the occluded markers can help to estimate the (few) isolated markers even if they are missing over a long period of time. However, the skeleton information must be known a priori in order to apply this method. [39] also takes advantage of the fact that the markers on a limb have fixed inter-marker pairwise distances. Thus, in the case where a marker is missing, its position can be recovered through the distance constraints imposed by the markers of the same limb. This approach may become ineffective when all or a significant number of

markers are missing so that the limb is unable to be inferred from the available neighboring markers. Ringer and Lasenby, [28], also present an automatic method to identify indistinguishable markers based on cliques<sup>1</sup>. However, this requires an off-line procedure in order to determine marker cliques and parameters of the skeletal structure.

In [40], 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. Although it is a real-time processing method, the knowledge of skeleton information eliminates its use more generally. Chai and Hodgins, [41], present a method that uses the neighbouring markers to estimate the missing marker in the current frame. They proposed 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 the model. This method is effective but the set of controls signals, i.e. markers, and the skeleton information must be known before the session.

Recently, Liu et al., in [42, 43], presented a piecewise linear approach to estimating the 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. However, the pre-training session and the classifier limit the approach to off-line applications.

Some other methods exist which deal with the cases where there is inadequate data and the centre of rotation is calculated with the use of two markers only. However this approach is effective only in cases where data from only one marker is missing each time, and is therefore not very realistic. Also, in this approach, information on the rotation around the axis joining the two remaining markers on the limb cannot be determined.

The aim of predicting the position of the occluded markers is for the calculation of the CoR between the two limbs. Instead of predicting the position of the occluded markers and then calculating the CoR, might a direct prediction of the CoR be possible? Unfortunately, this is not feasible because calculating the CoR when all the markers are active requires information regarding to the rotors between the previous and the current frames,  $R_x^{k-1,k}$ . This rotor is used for the calculation of the rotor  $R_x^{1,k}$ , which gives the rotation between the current frame and a reference frame (see figure 3.3). Therefore, a method based on the prediction of the occluded markers will be more practicable.

## 4.2. Kalman Filter

The Kalman filter [44] is a commonly used method for tracking in many different areas, such as autonomous or assisted navigation, interactive computer graph-

---

<sup>1</sup>Markers in a clique have constant distances between each other

ics and motion prediction. The simplicity and the robust nature of the filter itself makes it very popular and practical for almost every design of prediction algorithm.

The process model to update the *state* of the Kalman filter is given by (4.1), where the state  $\mathbf{x}_k$  at frame  $k$  is obtained from the state at frame  $k - 1$ ;

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (4.1)$$

where  $A$  is the state transition model,  $B$  is the control input model,  $\mathbf{u}_{k-1}$  is the control vector and  $\mathbf{w}_{k-1}$  is the process noise. The measured data  $\mathbf{Z}_k$  is

$$\mathbf{Z}_k = H\mathbf{x}_k + \mathbf{v}_k \quad (4.2)$$

where  $H$  is the *observation model* and  $\mathbf{v}_k$  is the observation noise.  $\mathbf{w}$  and  $\mathbf{v}$  are assumed to be zero mean multivariate normal with covariance  $Q$  and  $R$  respectively.

The predicted state  $\mathbf{y}_k$  and its error  $E_k$  can be written as

$$\mathbf{y}_k = A\hat{\mathbf{x}}_{k-1} + B\mathbf{u}_{k-1} \quad E_k = AP_{k-1}A^T + Q \quad (4.3)$$

where  $\hat{\mathbf{x}}$  refers to the *estimate* and  $P$  is the covariance of the state estimate.

The *Kalman gain* between actual and predicted observations is:

$$K_k = E_k H^T (H E_k H^T + R)^{-1} \quad (4.4)$$

Thus given an estimate  $\hat{\mathbf{x}}_{k-1}$  at  $k - 1$ , the time update predicts the state value at frame  $k$ . The measurement update then adjusts this prediction based on the new  $\mathbf{y}_k$ . The estimate of the new state is

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + K_k (\mathbf{Z}_k - H\mathbf{y}_k) \quad (4.5)$$

The innovation gain  $K_k$  is chosen to minimise the steady-state covariance of the estimation error given the state noise covariance  $p(\mathbf{w}) \sim N(0, Q)$  and the observation noise covariance  $p(\mathbf{v}) \sim N(0, R)$ . The measurement noise variance  $R$  might change with each time step or measurement. Also, the process noise variance  $Q$  might change in order to adjust to different dynamics. However, in this work it is assumed that they stay constant during the filter operation. Finally, the error covariance matrix of the updated estimate is;

$$P_k = (I - K_k H) E_k \quad (4.6)$$

In this chapter we will use a constant velocity model. Our goal is to build a model that predicts the current state using previous states. The equation which deals with the constant velocity model and expresses the next possible position of the marker can be formulated as given below:

$$\mathbf{y}_k = \mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1} dk \quad (4.7)$$

where  $\mathbf{x}_i$  and  $\dot{\mathbf{x}}_i$  are the position and velocity in frame  $i$ , and  $dk$  is the time step. The assumption made here is that the acceleration remains stable during two frames in a row.

Thus, equation 4.3, which express the predicted state, can be solved in a matrix model as listed below:

$$\begin{bmatrix} \mathbf{x}_k \\ \dot{\mathbf{x}}_k \end{bmatrix} = \begin{bmatrix} I_3 & dkI_3 \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \dot{\mathbf{x}}_{k-1} \end{bmatrix} + B\mathbf{u}_{k-1} \quad (4.8)$$

It is important here to remember that each marker position comprises  $x$ ,  $y$  and  $z$  coordinates. Assuming that  $B = 0$  and  $dk = 1$ , matrix  $A$  which updates the state can be expressed as  $A = \begin{bmatrix} I_3 & I_3 \\ 0 & I_3 \end{bmatrix}$ .

### 4.3. Applying Constraints

In this section we will apply constraints to handle cases where the occlusion occurs over a significant number of frames. To cope with such cases, the main idea is to implement a tracker that uses information not only from the previous frames, but also from the current position of neighbouring visible markers. The rigid body has some important properties useful for the estimation of occluded markers. The markers are located at strategic points on the articulated body and have their inter-marker distances constant.

A very important parameter for the accurate operation of the Kalman filter is the observation vector. The observation vector,  $\mathbf{Z}_k$ , gives the observed position of the tracked marker when available, otherwise it represents estimated position. We assume three markers on each limb. In the presence of noise the observation vector is updated as given below for 5 different scenarios. These methods can take care of cases where the velocity, the acceleration and/or the direction of the non-visible marker is rapidly changing during the occlusion period.

#### 4.3.1. All markers are visible on a given limb

Where all markers are visible on a given limb, then:

$$\mathbf{Z}_k = H\mathbf{x}_k + \mathbf{v}_k \quad (4.9)$$

where  $\mathbf{x}_k$  is the current state of a tracked marker on the limb. In this case  $H$  is the identity and  $R$  is determined empirically. Many factors contribute to marker noise, and hence  $R$ , including optical measurement noise, miscalibration of the optical systems, reflection, motion of markers relative to the skin and motion of the skin relative to the rigid body (underlying bone).

### 4.3.2. One missing marker on a limb segment

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

$$\mathbf{Z}_k = H\hat{\mathbf{x}}_1^k + \mathbf{v}_k \quad (4.10)$$

where  $\hat{\mathbf{x}}_1^k$  is the estimated position of the occluded marker  $m_1$  in frame  $k$ .  $\hat{\mathbf{x}}_1^k$  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 the vectors between marker  $m_1$  and markers  $m_2, m_3$  in frame  $k-1$  respectively. These vectors are given by (see figure Fig 4.1):

$$\mathbf{D}_{i,j}^{k-1} = \mathbf{x}_j^{k-1} - \mathbf{x}_i^{k-1} \quad (4.11)$$

Thereafter, these vectors are rotated as

$$\hat{\mathbf{D}}_{i,j}^k = R^{k-2,k-1}\mathbf{D}_{i,j}^{k-1}\tilde{R}^{k-2,k-1} \quad (4.12)$$

where  $R^{p,q}$  is the rotor for the rotation between frames  $p$  to  $q$ , assuming that the rotation of the markers between two consecutive frames remains constant. 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;

$$\tilde{\mathbf{x}}_1^k = \frac{(\mathbf{x}_2^k - \hat{\mathbf{D}}_{1,2}^k) + (\mathbf{x}_3^k - \hat{\mathbf{D}}_{1,3}^k)}{2} \quad (4.13)$$

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  $\tilde{\mathbf{x}}_1^k$ . The solution to this problem is given in [45]. Figure 4.1 illustrates this process.

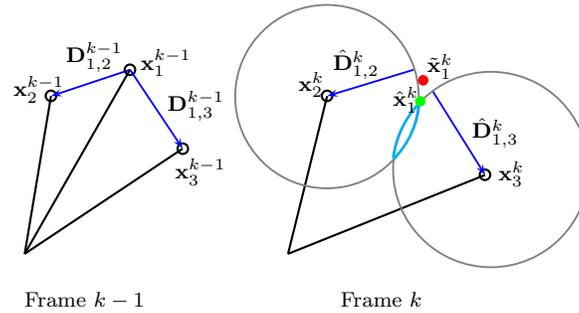


Figure 4.1.: The observation vector in the case of 2 visible markers. The red dot,  $\tilde{\mathbf{x}}_1^k$ , represents the average value as given in equation 4.13. The green dot,  $\hat{\mathbf{x}}_1^k$ , is the point on the intersection of the 2 spheres which is closest to  $\tilde{\mathbf{x}}_1^k$ .

### 4.3.3. Two missing markers on a limb segment

In the case of only one marker ( $m_2$ ) visible on a given limb we again have;

$$\mathbf{Z}_k = H\hat{\mathbf{x}}_j^k + \mathbf{v}_k \quad (4.14)$$

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:

$$\hat{\mathbf{x}}_j^k = \mathbf{x}_2^k - \hat{\mathbf{D}}_{j,2}^k \quad (4.15)$$

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. In this case, we are using the constant velocity assumption as we cannot estimate the rotation. In that case, it is assumed that the rotation around the axis joining the two remaining markers on the limb does not exist. Figure 4.2 demonstrates the above.

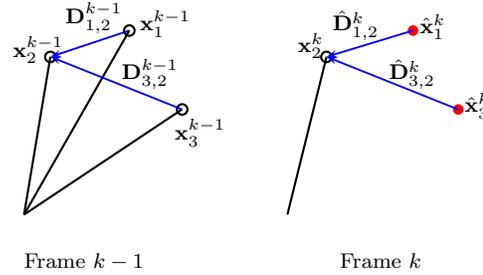


Figure 4.2.: The observation vector in the case of only one visible marker. The red dots,  $\hat{\mathbf{x}}_j^k$ , where  $j = \{1, 3\}$  represent the estimated position of the missing marker  $m_j$  as given in equation 4.15.

### 4.3.4. 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 (as in figure 3.2) 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, the missing marker positions can be calculated using the CoR estimate,  $\hat{\mathbf{C}}_k$ , as calculated in section 3.4.2, equation 3.35. In that case the observation vector of the Kalman filter is updated as:

$$\mathbf{Z}_k = H\hat{\mathbf{x}}_j^k + \mathbf{v}_k \quad (4.16)$$

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;

$$\hat{\mathbf{x}}_j^k = \hat{\mathbf{C}}_k + \hat{\mathbf{D}}_{j,c}^k \quad (4.17)$$

where  $\hat{\mathbf{D}}_{j,c}^k$  is an estimate of the vector between marker  $m_i$  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 4.3 illustrates the above.

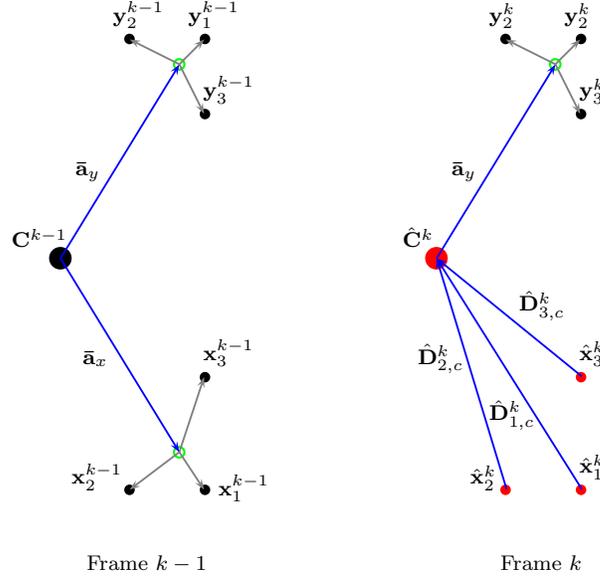


Figure 4.3.: The estimation procedure when all markers on a single limb segment are occluded. The red dots represents the estimated position of the CoR,  $\hat{\mathbf{C}}^k = \mathbf{y}_i^k - R_y^k \bar{\mathbf{a}}_y \tilde{R}_y^k$  where  $i = \{1, 2, 3\}$ , and the estimated marker positions of the 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.

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

$$\mathbf{Z}_k = H \hat{\mathbf{x}}^k + \mathbf{v}_k \quad (4.18)$$

where  $\hat{\mathbf{x}}^k$  is equal to  $\hat{\mathbf{x}}^k = R^{k-2,k-1} \mathbf{x}^{k-1} \tilde{R}^{k-2,k-1}$ .

A different approach for estimating the missing markers when all markers are occluded uses a constant velocity model. It is assumed that the velocity of the occluded limb remains stable; thus the observation vector is formulated as:

$$\mathbf{Z}_k = H \mathbf{x}_1^{k-1} + \mathbf{U}_k \quad (4.19)$$

where  $\mathbf{U}_t$  is equal to  $\mathbf{U}_t = \frac{(\mathbf{x}_1^{t-1} - \mathbf{x}_1^{t-2}) + (\mathbf{x}_1^{t-1} - \hat{\mathbf{x}}_1^{t-1})}{2}$  and  $\hat{\mathbf{x}}_1^{t-1}$  is the predicted state using the Kalman filter on the previous frame. The  $\mathbf{x}_1^{t-1}$  and  $\mathbf{x}_1^{t-2}$  are the real positions of the tracked marker in the previous frame and two frames ago, respectively. However, this method is not as efficient as the constant rotor approach. Using the constant rotor approach, we ensure that on each marker the same rotation has been applied; thus, markers on the same limb will obey the rigid body assumption. When the constant velocity method is used, each marker has its own trajectory “violating” the skeletal model of the tracked limb. This has a continuous addition of noise and hence, the calculated CoR has a bigger error.

#### 4.3.5. Markers visible in only one camera

The motion capture system also provides us with additional information which could be used for prediction of missing marker locations. Each marker can be reconstructed by the motion capture system if it is visible in at least two cameras. It is often the case that some markers are visible in only one 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 the line  $L_1$ , we may be able to obtain a more accurate estimate of the position of the relevant marker. This position,  $\hat{\mathbf{x}}_1^k$ , corresponds to the projection from the point  $\hat{\mathbf{x}}_1^k$  onto the line  $L_1$ , as in figure 4.4. 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.

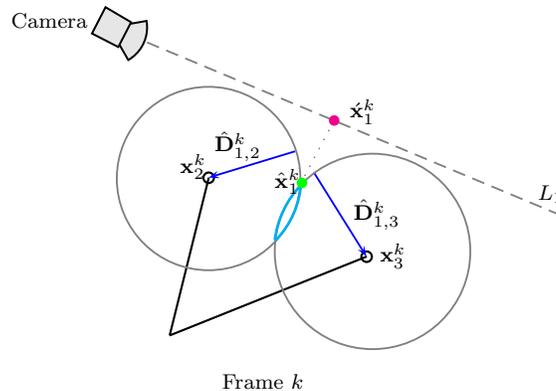


Figure 4.4.: The observation vector in the case of 2 visible markers and one marker visible only by one camera. The magenta dot,  $\hat{\mathbf{x}}_1^k$ , is now used for the calculation of the observation vector,  $\mathbf{Z}_k = H\hat{\mathbf{x}}^k + \mathbf{v}_k$ .

Next we present an alternative method for estimating the missing markers

when the relevant marker is visible in one camera. In the case where two markers are visible, instead of projecting  $\hat{\mathbf{x}}_j^k$  onto the line  $L_1$ , the missing marker,  $\hat{\mathbf{x}}_j^k$ , can be estimated by finding the nearest point on the line  $L_1$  from the circle created after the intersection of the two spheres. In the case where one marker is visible, then the missing marker would be the point on line  $L_1$  having minimum distance from sphere  $\Sigma_2$  where  $\Sigma_2$  has as its centre the position of the visible marker,  $\mathbf{x}_2^k$ , and its radii is the distance  $|\hat{\mathbf{D}}_{j,2}^k|$ , where  $m_j$  is the missing marker. These methods are presented in detail in [45]. One might suppose that these methods will return more accurate results. However, using the information coming from the rotated distance,  $\hat{\mathbf{D}}_{i,j}^k$ , using the previous frame, helps us to locate a more reliable position for the missing marker. By ignoring this information, even if the estimated position meets all the conditions (the circle ensures inter-marker constant distance and that the point will be on the line), the algorithm is not as reliable as the methods proposed previously.

# 5

## Filtering extension using dynamic models and rotation prediction

---

### 5.1. Dynamic Model for Target Tracking

#### 5.1.1. Introduction

Most tracking problems require a dynamic model for accurate estimation of the trajectory of a maneuverable object. The key for an efficient target tracking algorithm is being able to extract information about the target's state from the observations. Most tracking algorithms are model based while knowledge of target motion is available, observation data may be limited. Hence, instead of using a constant velocity model, as in equation 4.8, for target tracking, it will be more efficient if we use a more sophisticated model, which take accounts the velocity and the acceleration of the target state.

During the last four decades various mathematical models of target motions have been developed [46]. Singer, [47, 48] 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 been proposed having a constant or variable acceleration. For a more detailed overview on these types of modelling methods, refer to [46].

### 5.1.2. Dynamic State-Space Model

The general state-space model can be generally divided into two models, the state transition model and the state measurement model

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}) \quad (5.1)$$

$$p(\mathbf{y}_t | \mathbf{x}_t) \quad (5.2)$$

where  $\mathbf{x}_t \in \mathbb{R}^{n_x}$  denotes the states of the system at time  $t$  and  $\mathbf{y}_t \in \mathbb{R}^{n_y}$  the observations. The states follow a first order Markov process and the observations are assumed to be independent given the states. Therefore, the dynamic model can be expressed as follows

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) \quad \mathbf{y}_t = \mathbf{h}(\mathbf{u}_t, \mathbf{x}_t, \mathbf{n}_t) \quad (5.3)$$

where,  $\mathbf{y}_t \in \mathbb{R}^{n_y}$  is the output observations,  $\mathbf{u}_t \in \mathbb{R}^{n_u}$  is the input observation,  $\mathbf{x}_t \in \mathbb{R}^{n_x}$  is the state of the system,  $\mathbf{v}_t \in \mathbb{R}^{n_v}$  is the process noise and  $\mathbf{n}_t \in \mathbb{R}^{n_n}$  the measurement noise. The mapping  $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \mapsto \mathbb{R}^{n_x}$  and  $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \mapsto \mathbb{R}^{n_y}$  represent the deterministic process and measurement models.

### 5.1.3. Constant Turn Model

A simple and popular model used to track the motion of a target is the nearly constant turn (NCT) model [46], which assumes that the target moves with nearly constant speed and turn rate. The altitude changes are most often modeled independently by a nearly constant velocity model (CV) or a random walk model along the  $\mathbf{z}$  direction, leading to an acceptable accuracy in practice.

It is easily proved that a constant speed (i.e  $\dot{\nu} = 0$ ) motion corresponds to  $\mathbf{a} \cdot \mathbf{v} = 0$ , where  $\mathbf{a}$  is the target acceleration vector and  $\mathbf{v}$  is the target velocity vector and is described equivalently by

$$\mathbf{a} = \boldsymbol{\Omega} \times \mathbf{v} \quad (5.4)$$

where  $\boldsymbol{\Omega}$  is the angular velocity vector of the target and is equal to

$$\boldsymbol{\Omega} = \frac{\mathbf{v} \times \mathbf{a}}{\nu^2} \quad (5.5)$$

and  $\nu = \|\mathbf{v}\|$  is the target speed. Assuming  $\dot{\boldsymbol{\Omega}} = 0$ , it follows that

$$\dot{\mathbf{a}} = \boldsymbol{\Omega} \times \mathbf{a} = \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{v}) = (\boldsymbol{\Omega} \cdot \mathbf{v})\boldsymbol{\Omega} - (\boldsymbol{\Omega} \cdot \boldsymbol{\Omega})\mathbf{v} = -\omega^2 \mathbf{v} \quad (5.6)$$

where the turn rate  $\omega$  is given by

$$\omega \triangleq \|\boldsymbol{\Omega}\| = \frac{\|\mathbf{v} \times \mathbf{a}\|}{\nu^2} = \frac{\|\mathbf{v}\| \|\mathbf{a}\|}{\nu^2} = \frac{a}{\nu} \quad (5.7)$$

Therefore, the NCT motion can be modeled by a second-order Markov process

$$\dot{\mathbf{a}} = -\omega^2 \mathbf{v} + \mathbf{w} \quad (5.8)$$

where  $\mathbf{w}$  is white noise.

In [46], it is proven that the target equations of the 3D discrete motion of this model in the Cartesian coordinates with state  $\mathbf{x}_k = [x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k, z_k, \dot{z}_k, \ddot{z}_k]^T$  may be represented by

$$\mathbf{x}_k = \text{diag}[F_x(\omega_x), F_y(\omega_y), F_z(\omega_z)] \mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (5.9)$$

where  $F_i(\omega_i)$  for each component  $i = x, y, z$  is equal to:

$$F_i(\omega_i) = \begin{bmatrix} 1 & \frac{\sin(\omega_i T)}{\omega_i} & \frac{1 - \cos(\omega_i T)}{\omega_i^2} \\ 0 & \cos(\omega_i T) & \frac{\sin(\omega_i T)}{\omega_i} \\ 0 & -\omega_i \sin(\omega_i T) & \cos(\omega_i T) \end{bmatrix} \quad \text{for } i = x, y, z \quad (5.10)$$

$$\text{cov}(\mathbf{w}_{k-1}) = \text{diag}[S_x Q_x(\omega_x), S_y Q_y(\omega_y), S_z Q_z(\omega_z)] \quad (5.11)$$

where  $S_x$ ,  $S_y$  and  $S_z$  are the power spectral density of the white noise  $\mathbf{w}$  of each component  $x, y, z$  and  $Q_i(\omega_i)$  for  $i = x, y, z$  is equal to

$$Q_i(\omega_i) = \begin{bmatrix} \frac{6\omega_i T - 8\sin(\omega_i T) + \sin(2\omega_i T)}{4\omega_i^5} & \frac{2\sin^4(\omega_i T/2)}{\omega_i^4} & \frac{-2\omega_i T + 4\sin(\omega_i T) - \sin(2\omega_i T)}{4\omega_i^3} \\ \frac{2\sin^4(\omega_i T/2)}{\omega_i^4} & \frac{2\omega_i T - \sin(2\omega_i T)}{4\omega_i^3} & \frac{\sin^2(\omega_i T)}{2\omega_i^2} \\ \frac{-2\omega_i T + 4\sin(\omega_i T) - \sin(2\omega_i T)}{4\omega_i^3} & \frac{\sin^2(\omega_i T)}{2\omega_i^2} & \frac{2\omega_i T - \sin(2\omega_i T)}{4\omega_i^3} \end{bmatrix} \quad (5.12)$$

The motions in the  $x, y, z$  directions in this model are coupled only through the turn rate  $\omega_i$ . This model specifies a constant-turn motion defined by the velocity and acceleration vectors of each component separately. For implementing such a model, the  $\omega_i$  must be known. In some cases,  $\omega$  can be computed using certain techniques, as reported in [46].

#### 5.1.4. Variable Turn Model

The above model was based on the constant-speed condition and the constant turn-rate assumption which therefore restricts the variety of possible maneuvers. In this section, a more sophisticated and more accurate model is examined which

does not assume that markers have constant velocity and turn-rate. This is the variable turn model (VTM). Figure 5.1 compares target prediction with variable acceleration using the VTM model against a linear prediction.

Asseo and Ardila in [49] describe a general motion of a rigid body in space. They show that, under the orthogonal velocity condition  $\boldsymbol{\Omega} \perp \mathbf{v}$  ( $\boldsymbol{\Omega}$  is given by eq. 5.5), the target acceleration can be expressed as a second-order Markov process with state dependence

$$\dot{\mathbf{a}} = -2\alpha\mathbf{a} - (2\alpha^2 + \omega^2)\mathbf{v} + \mathbf{w} \quad \mathbf{w} = \frac{d^2\mathbf{v}}{dt^2} + \dot{\boldsymbol{\Omega}} \times \mathbf{v} \quad (5.13)$$

where

$$\omega \triangleq \|\boldsymbol{\Omega}\| = \frac{\|\mathbf{v} \times \mathbf{a}\|}{\nu^2} \quad \alpha = -\frac{\mathbf{v} \cdot \mathbf{a}}{\nu^2} \quad (5.14)$$

The target equations of the 3D discrete motion of the VTM model in Cartesian coordinates with state  $\mathbf{x}_k = [x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k, z_k, \dot{z}_k, \ddot{z}_k]^T$  is now represented by

$$\mathbf{x}_k = \text{diag}[F_x(\omega_{c,x}, \alpha_x), F_y(\omega_{c,y}, \alpha_y), F_z(\omega_{c,z}, \alpha_z)] \mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

where  $F_i(\omega_{c,i}, \alpha_i)$  for each component  $i = x, y, z$  is given by:

$$F_i(\omega_{c,i}, \alpha_i) = \begin{bmatrix} 1 & \frac{2\alpha_i\omega_{c,i} - e^{-\alpha_i T}(2\alpha_i\omega_{c,i}\cos(\omega_{c,i}T) + (\alpha_i^2 - \omega_{c,i}^2)\sin(\omega_{c,i}T))}{\omega_{c,i}(\alpha_i^2 + \omega_{c,i}^2)} & \frac{\omega_{c,i} - e^{-\alpha_i T}(\omega_{c,i}\cos(\omega_{c,i}T) + \alpha_i\sin(\omega_{c,i}T))}{\omega_{c,i}(\alpha_i^2 + \omega_{c,i}^2)} \\ 0 & \frac{e^{-\alpha_i T}(\omega_{c,i}\cos(\omega_{c,i}T) + \alpha_i\sin(\omega_{c,i}T))}{\omega_{c,i}} & \frac{e^{-\alpha_i T}\sin(\omega_{c,i}T)}{\omega_{c,i}} \\ 0 & -\frac{(\alpha_i^2 + \omega_{c,i}^2)e^{-\alpha_i T}\sin(\omega_{c,i}T)}{\omega_{c,i}} & \frac{e^{-\alpha_i T}(\omega_{c,i}\cos(\omega_{c,i}T) + \alpha_i\sin(\omega_{c,i}T))}{\omega_{c,i}} \end{bmatrix}$$

and

$$Q = \text{cov}(\mathbf{w}_{k-1}) = \text{diag}[S_x Q_x(\omega_{c,x}, \alpha_x), S_y Q_y(\omega_{c,y}, \alpha_y), S_z Q_z(\omega_{c,z}, \alpha_z)] \quad (5.15)$$

where  $\omega_{c,i}^2 = \alpha_i^2 + \omega_i^2$ ,  $S_x$ ,  $S_y$  and  $S_z$  are the power spectral density of the white noise  $\mathbf{w}$  of each component  $x, y, z$ .  $Q_i(\omega_{c,i}, \alpha_i)$  for  $i = x, y, z$  is given by:

$$q_{11} = \frac{A + B + C}{4\alpha_i\omega_{c,i}^2(\alpha_i^2 + \omega_{c,i}^2)^3}$$

$$A = e^{-2\alpha_i T}[(\alpha_i^2 - 3\omega_{c,i}^2)c + (\omega_{c,i}^2 - 3\alpha_i^2)s - (\alpha_i^2 + \omega_{c,i}^2)^2]$$

$$B = 8e^{-\alpha_i T}\alpha_i\omega_{c,i}[2\alpha_i\omega_{c,i}c_0 + (\alpha_i^2 - \omega_{c,i}^2)s_0]$$

$$C = \alpha_i^2\omega_{c,i}^2(4\alpha_i T - 11) + \omega_{c,i}^4(1 + 4\alpha_i T)$$

(5.16)

$$q_{12} = \frac{e^{-2\alpha_i T} (\omega_{c,i} c_0 + \alpha_i s_0 - e^{-\alpha_i T} \omega_{c,i})^2}{2\omega_{c,i}^2 (\alpha_i^2 + \omega_{c,i}^2)^2}$$

$$q_{13} = \frac{e^{-2\alpha_i T} (c - s - \alpha_i + \omega_{c,i}^2) + 4e^{-\alpha_i T} \alpha_i \omega_{c,i} s_0 - \omega_{c,i}^2}{4\alpha_i \omega_{c,i}^2 (\alpha_i^2 + \omega_{c,i}^2)^2}$$

$$q_{22} = \frac{e^{-2\alpha_i T} (c - s - \alpha_i - \omega_{c,i}^2) + \omega_{c,i}^2}{4\alpha_i \omega_{c,i}^2 (\alpha_i^2 + \omega_{c,i}^2)^2}$$

$$q_{23} = \frac{e^{-2\alpha_i T} s_0^2}{2\omega_{c,i}^2}$$

$$q_{33} = \frac{e^{-2\alpha_i T} (c + s - \alpha_i - \omega_{c,i}^2) + \omega_{c,i}^2}{4\alpha_i \omega_{c,i}^2}$$

$$q_{12} = q_{13} = q_{23} = 0$$

$$c = \alpha_i^2 \cos(2\omega_{c,i} T), \quad s = \alpha_i \omega_{c,i} \sin(2\omega_{c,i} T)$$

$$c_0 = \cos(\omega_{c,i} T), \quad s_0 = \sin(\omega_{c,i} T)$$

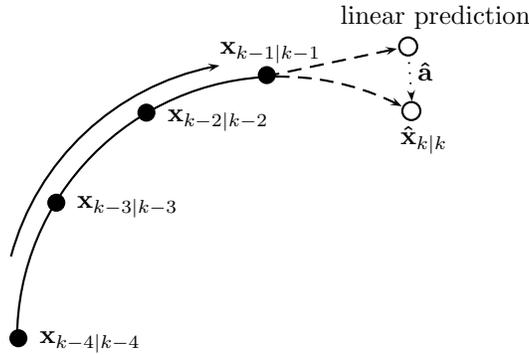


Figure 5.1.: Target prediction with variable acceleration using the VTM model compared to the linear prediction.

A more detailed look at the marker data, indicates that the velocity noise is huge due to the marker position noise. Many factors contribute to marker position noise such as optical measurement noise, miscalibration of the optical systems, reflection, 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 noise, having only little information related to that target motion. Hence, the dynamic model proposed above becomes useless. In order to overcome this problem, we applied a real-time median filter to the target velocity (see figure

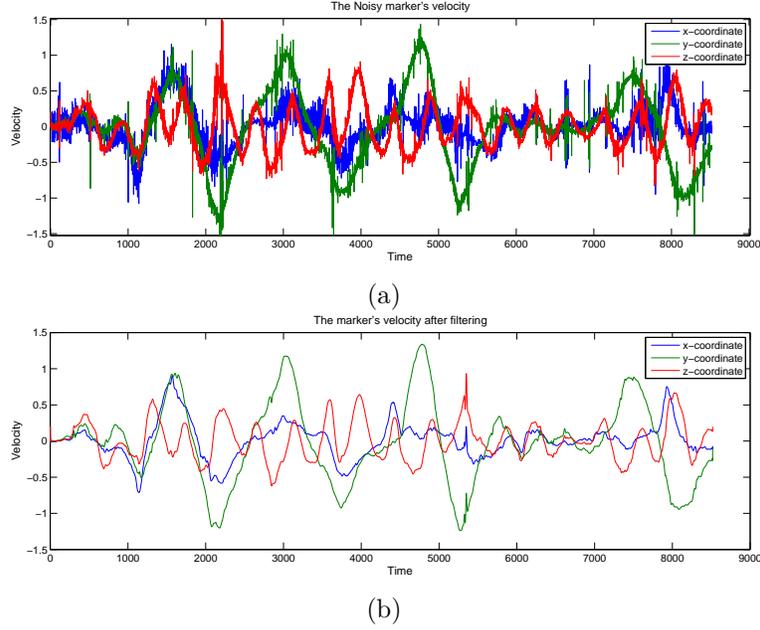


Figure 5.2.: **(a)** The noisy marker's velocity and **(b)** the same velocity after filtering using a median filter.

5.2). Having a smoothed velocity, the acceleration in eq. 5.14 can be calculated with more accuracy and therefore, the proposed dynamic model can be used without problems (see figure 5.3).

## 5.2. Unscented Transform

Kalman filtering has been extensively used for real time estimation of linear dynamic systems. However, the traditional Kalman filter 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) [50] is a minimum mean-square-error (MMSE) estimator which extends the scope of the Kalman filter to nonlinear optimal filtering problems by forming a Gaussian approximation to the joint distribution of state  $\mathbf{x}$  and measurement  $\mathbf{y}$  using a Taylor series based transformation. Nevertheless, according to [51], the EKF has a few serious drawbacks which should be considered when it is used. An EKF implementation is complex (Jacobian and Hessian matrices with second order filters), difficult to tune, and only reliable for systems that are almost linear on the time scale of the updates. Most of these difficulties arise from its use of linearisation. The EKF only uses the first order terms of the Taylor series expansion of the nonlinear functions, hence it often introduces large errors in the estimated statistics of the posterior distributions of the states. To overcome this limitation, the unscented transformation (UT) was developed by Julier and Uhlmann [52]

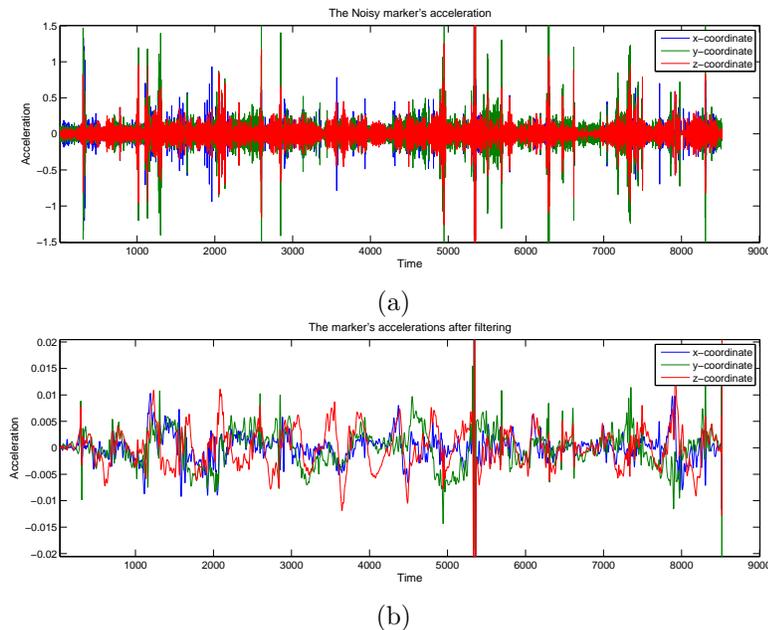


Figure 5.3.: **(a)** The acceleration using the noisy marker's velocity and **(b)** the same acceleration using the filtered velocity.

as a method to propagate mean and covariance information through nonlinear transformations. The UT is more accurate, easier to implement, and uses the same order of calculations as linearisation.

### 5.2.1. Unscaled Unscented Transform

The Unscented Transform can be used for forming a Gaussian approximation to the joint distribution of random variables  $\mathbf{x}$  and  $\mathbf{y}$ . UT deterministically chooses a fixed number of '*sigma points*', which capture the desired moments (mean and covariance) of the original distribution of  $\mathbf{x}$  (details about *sigma points* can be found later). After that, it propagates the sigma points through the non-linear function  $\mathbf{g}$  and estimates the moments of the transformed variable from them. The UT has the advantage of capturing the higher order moments caused by the non-linear transform and is less complex compared to the EKF since the Jacobian and Hessian matrices are not needed. Also, the predictions made by the family of UTs are less error-prone compared to any other real-time prediction algorithm.

The unscented transformation (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation and builds on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function [52]. Consider propagating a  $n_x$ -dimensional random variable  $\mathbf{x}$  through an arbitrary nonlinear function  $\mathbf{g} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$  to generate  $\mathbf{y}$ ,

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \quad (5.17)$$

Assume  $\mathbf{x}$  has mean  $\bar{\mathbf{x}}$  and covariance  $\mathbf{P}_x$ . To calculate the statistics (first two moments) of  $\mathbf{y}$  using the UT, we proceed as follows: 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 they completely capture the true mean and covariance of the random variable  $\mathbf{x}$ . A selection scheme that satisfies this requirement is

$$\begin{aligned} \mathcal{X}_0 &= \bar{\mathbf{x}} & W_0 &= \kappa / (n_x + \kappa) & i &= 0 \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \left( \sqrt{(n_x + \kappa) \mathbf{P}_x} \right)_i & W_i &= 1 / \{2(n_x + \kappa)\} & i &= 1, \dots, n_x \\ \mathcal{X}_i &= \bar{\mathbf{x}} - \left( \sqrt{(n_x + \kappa) \mathbf{P}_x} \right)_i & W_i &= 1 / \{2(n_x + \kappa)\} & i &= n_x + 1, \dots, 2n_x \end{aligned} \quad (5.18)$$

where  $\kappa$  is a scaling parameter and  $\left( \sqrt{(n_x + \kappa) \mathbf{P}_x} \right)_i$  is the  $i$ th row or column of the matrix square root of  $(n_x + \kappa) \mathbf{P}_x$ .  $W_i$  is the weight associated with the  $i$ th point such that  $\sum_{i=0}^{2n_x} W_i = 1$ . Each sigma point is now propagated through the nonlinear function

$$\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i) \quad i = 0, \dots, 2n_x \quad (5.19)$$

and the estimated mean and covariance of  $\mathbf{y}$  are computed as follows

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n_x} W_i \mathcal{Y}_i \quad (5.20)$$

$$\mathbf{P}_y = \sum_{i=0}^{2n_x} W_i (\mathcal{Y}_i - \bar{\mathbf{y}}) (\mathcal{Y}_i - \bar{\mathbf{y}})^T \quad (5.21)$$

These estimates of the mean and covariance are accurate to the second order (third order for Gaussian priors) of the Taylor series expansion of  $\mathbf{g}(\mathbf{x})$  for any nonlinear function. Errors are introduced in the third and higher order moments but are scaled by the choice of the parameter  $\kappa$ . In comparison, the EKF only calculates the posterior mean and covariance accurately to the first order with all higher order moments truncated. For a detailed proof of this, see [53].

The sigma point selection scheme used in the UT has the property that as the dimension of the state-space increases, the radius of the sphere that bounds all the sigma points increases as well. Even though the mean and covariance of the prior distribution are still captured correctly, it does so at the cost of sampling non-local effects. If the nonlinearities in question are very severe, this can lead to significant difficulties. In order to address this problem, the sigma points can be scaled towards or away from the mean of the prior distribution by a proper choice of  $\kappa$ . The distance of the  $i$ th sigma point from  $\bar{\mathbf{x}}$ ,  $|\mathcal{X}_i - \bar{\mathbf{x}}|$ , is proportional to  $\sqrt{(n_x + \kappa)}$ . When  $\kappa = 0$ , the distance is proportional to  $\sqrt{n_x}$ . When  $\kappa > 0$  the points are scaled further from  $\bar{\mathbf{x}}$  and when  $\kappa < 0$  the points are scaled towards  $\bar{\mathbf{x}}$ . For the special case of  $\kappa = 3 - n_x$ , the desired dimensional scaling invariance

is achieved by canceling the effect of  $n_x$ . However, when  $\kappa = 3 - n_x < 0$  and the weight  $W_0 < 0$ , the calculated covariance can be non-positive semi-definite. The scaled unscented transformation was developed to address this problem [54].

### 5.2.2. Scaled Unscented Transform

The scaled unscented transformation (SUT) replaces the original set of sigma points with a transformed set given by

$$\mathcal{X}'_i = \mathcal{X}_0 + \alpha (\mathcal{X}_i - \mathcal{X}_0) \quad i = 0, \dots, 2n_x \quad (5.22)$$

where  $\alpha$  is a positive scaling parameter which can be made arbitrarily small to minimise higher order effects. This formulation gives an extra degree of freedom to control the scaling of the sigma points without causing the resulting covariance to possibly become non-positive semidefinite. This is achieved by applying the UT to an auxiliary random variable propagation problem which is related to the original nonlinear model of equation 5.17 by:

$$\mathbf{z} = \mathbf{g}'(\mathbf{x}) = \frac{\mathbf{g}[\bar{\mathbf{x}} + \alpha(\mathbf{x} - \bar{\mathbf{x}})] - \mathbf{g}(\bar{\mathbf{x}})}{\alpha^2} + \mathbf{g}(\bar{\mathbf{x}}) \quad (5.23)$$

The Taylor series expansion of  $\mathbf{z}$  and  $\mathbf{P}_z$  agrees with that of  $\mathbf{y}$  and  $\mathbf{P}_y$  exactly up to the second order, with the higher order terms scaling geometrically with a common ratio of  $\alpha$ . The same second order accuracy of the normal UT is thus retained with a controllable scaling of the higher order errors by a proper choice of  $\alpha$ . The auxiliary random variable formulation of the SUT is identical to applying the original UT on a pre-scaled set of sigma points [54]. A set of sigma points  $\mathcal{S} = \{\mathbf{W}, \mathcal{X}\}$  is calculated using equation 5.18 and then transformed into the scaled set  $\mathcal{S}' = \{\mathbf{W}', \mathcal{X}'\}$  by

$$\begin{aligned} \mathcal{X}'_i &= \mathcal{X}_0 + \alpha(\mathcal{X}_i - \mathcal{X}_0) \\ W'_i &= \begin{cases} W_0/\alpha^2 + (1 - 1/\alpha^2), & i = 0 \\ W_i/\alpha^2, & i \neq 0 \end{cases} \end{aligned} \quad (5.24)$$

where  $\alpha$  is the new sigma point scaling parameter. The sigma point selection and scaling can also be combined into a single step (thereby reducing the number of calculations) by setting

$$\lambda = \alpha^2(n_x + \kappa) - n_x \quad (5.25)$$

and selecting the sigma point set by:

$$\begin{aligned}
\mathcal{X}_0 &= \bar{\mathbf{x}} & i &= 0 \\
\mathcal{X}_i &= \bar{\mathbf{x}} + \left( \sqrt{(n_x + \lambda) \mathbf{P}_x} \right)_i & i &= 1, \dots, n_x \\
\mathcal{X}_i &= \bar{\mathbf{x}} - \left( \sqrt{(n_x + \lambda) \mathbf{P}_x} \right)_i & i &= n_x + 1, \dots, 2n_x \\
W_0^{(m)} &= \lambda / (n_x + \lambda) & i &= 0 \\
W_0^{(c)} &= \lambda / (n_x + \lambda) + (1 - \alpha^2 + \beta) & i &= 0 \\
W_i^{(m)} &= W_i^{(c)} = 1 / \{2(n_x + \lambda)\} & i &= 1, \dots, 2n_x
\end{aligned} \tag{5.26}$$

The weighting on the 0th sigma point directly affects the magnitude of the errors in the fourth and higher order terms for symmetric prior distributions ([54]). A third parameter,  $\beta$ , is thus introduced which affects the weighting of the 0th sigma point for the calculation of the covariance. This allows for the minimisation of higher order errors if prior knowledge (i.e. kurtosis, etc.) of the distribution of  $\mathbf{x}$  is available.

The complete scaled unscented transformation is thus given as follow:

1. Choose the parameters  $\kappa$ ,  $\alpha$  and  $\beta$ . Choose  $\kappa \geq 0$  to guarantee positive semidefiniteness of the covariance matrix. The specific value of  $\kappa$  is not critical though, so a good default choice is  $\kappa = 0$ . Choose  $0 \leq \alpha \leq 1$  and  $\beta \geq 0$ .  $\alpha$  controls the *size* of the sigma point distribution and should ideally be a small number to avoid sampling non-local effects when the nonlinearities are strong.  $\beta$  is a non-negative weighting term which can be used to incorporate knowledge of the higher order moments of the distribution. For a Gaussian prior the optimal choice is  $\beta = 2$ . This parameter can also be used to control the error in the kurtosis which affects the ‘heaviness’ of the tails of the posterior distribution.
2. Calculate the set of  $2n_x + 1$  scaled sigma points and weights  $\mathcal{S} = \{\mathbf{W}, \mathcal{X}\}$  by setting  $\lambda = \alpha^2(n_x + \kappa) - n_x$  and using the combined selection/scaling scheme of equation 5.26. As mentioned earlier,  $n_x$  is the dimension of  $\mathbf{x}$ .
3. Propagate each sigma point through the nonlinear transformation

$$\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i) \quad i = 0, \dots, 2n_x \tag{5.27}$$

4. The mean  $\bar{\mathbf{y}}$  and covariance  $\mathbf{P}_y$  are computed as follows

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathcal{Y}_i \tag{5.28}$$

$$\mathbf{P}_y = \sum_{i=0}^{2n_x} W_i^{(c)} (\mathcal{Y}_i - \bar{\mathbf{y}}) (\mathcal{Y}_i - \bar{\mathbf{y}})^T \tag{5.29}$$

### 5.2.3. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a straightforward application of the scaled unscented transformation to recursive minimum mean-square-error (RMMSE) estimation [55], where the state random variable (RV) is redefined as the concatenation of the original state and noise variables at time  $k$ :  $\mathbf{x}_k^\alpha = [\mathbf{x}_k^T \mathbf{v}_k^T \mathbf{n}_k^T]^T$ . The SUT sigma point selection scheme is applied to this new augmented state RV to calculate the corresponding sigma matrix,  $\mathcal{X}_k^\alpha$ . The complete UKF algorithm that updates the mean  $\bar{\mathbf{x}}$  and covariance  $\mathbf{P}$  of the Gaussian approximation to the posterior distribution of the states is given by:

1. Initialise with:

$$\begin{aligned}\bar{\mathbf{x}}_0 &= E[\mathbf{x}_0] \\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T] \\ \bar{\mathbf{x}}_0^\alpha &= E[\mathbf{x}^\alpha] = [\mathbf{x}_0^T \ 0 \ 0]^T\end{aligned}\tag{5.30}$$

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

2. For  $k \in \{1, \dots, \infty\}$  then:

- a) Calculate sigma points:

$$\mathcal{X}_{k-1}^a = \left[ \bar{\mathbf{x}}_{k-1}^\alpha \ \bar{\mathbf{x}}_{k-1}^\alpha \pm \sqrt{(n_\alpha + \lambda) \mathbf{P}_{k-1}^\alpha} \right]\tag{5.31}$$

- b) Time update:

$$\begin{aligned}\mathcal{X}_{k|k-1}^x &= \mathbf{f}(\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^\nu) \\ \bar{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{2n_a} W_i^{(m)} \mathcal{X}_{i,k|k-1}^x \\ \mathbf{P}_{k|k-1} &= \sum_{i=0}^{2n_a} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^x - \bar{\mathbf{x}}_{k|k-1}] [\mathcal{X}_{i,k|k-1}^x - \bar{\mathbf{x}}_{k|k-1}]^T \\ \mathcal{Z}_{k|k-1} &= \mathbf{h}(\mathcal{X}_{k|k-1}^x, \mathcal{X}_{k-1}^n) \\ \bar{\mathbf{z}}_{k|k-1} &= \sum_{i=0}^{2n_a} W_i^{(m)} \mathcal{Z}_{i,k|k-1}^x\end{aligned}\tag{5.32}$$

c) Measurement update equations:

$$\begin{aligned}
\mathbf{P}_{\tilde{z}_k \tilde{z}_k} &= \sum_{i=0}^{2n_a} W_i^{(c)} [\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_{k|k-1}] [\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_{k|k-1}]^T \\
\mathbf{P}_{x_k z_k} &= \sum_{i=0}^{2n_a} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \bar{\mathbf{x}}_{k|k-1}] [\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_{k|k-1}]^T \\
\mathbf{K}_k &= \mathbf{P}_{x_k y_k} \mathbf{P}_{\tilde{y}_k \tilde{y}_k}^{-1} \\
\bar{\mathbf{x}}_k &= \bar{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{Z}_k - \bar{\mathbf{z}}_{k|k-1}) \\
\mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\tilde{z}_k \tilde{z}_k} \mathbf{K}_k^T
\end{aligned} \tag{5.33}$$

where,  $\mathbf{x}^\alpha = [\mathbf{x}^T \mathbf{v}^T \mathbf{n}^T]^T$ ,  $\mathcal{X}^\alpha = [(\mathcal{X}^x)^T (\mathcal{X}^\nu)^T (\mathcal{X}^n)^T]^T$ ,  $\lambda$  is the composite scaling parameter,  $n_\alpha = n_x + n_\nu + n_n$ ,  $\mathbf{Q}$  is the process noise covariance,  $\mathbf{R}$  is the measurement noise covariance,  $\mathbf{K}$  is the Kalman gain,  $W_i$  are the weights as calculated in equation 5.26 and  $\mathbf{Z}_k$  is the observation state.

Note that no explicit calculation of Jacobians or Hessians are necessary to implement this algorithm. The UKF requires computation of a matrix square root which can be implemented directly using a Cholesky factorization.

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 supposed to be a variable turn model (VTM) as described in section 5.1.4 with target velocity and acceleration the velocity and acceleration of the relevant marker. On the other hand, for the observation function we used a simple model which assumes that the rotation between two consecutive frames is constant. Thus,  $\mathcal{Z}_{k|k-1}$  in equation 5.32 will be equal to

$$\mathcal{Z}_{k|k-1} = R^{k-2,k-1} \mathcal{X}_{k|k-1}^x \tilde{R}^{k-2,k-1} \tag{5.34}$$

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

In the same order as in the simple Kalman Filter, similar marker constraints can be applied. Hence, the observation vector,  $\mathbf{Z}_k$ , can be calculated using the same methodology as in section 4.3.

### 5.3. Rotation Prediction

During the Kalman and Unscented Kalman filtering, when the constraints on the observation state have been applied, we assumed that the rotation between the previous and the current frame remains constant (see section 4.3). This actually adds a small error in the prediction state; the error is small because  $R_k$

is re-calculated and updated just after the marker prediction using the estimated positions. Nevertheless, a likely methodology of rotation prediction will be useful for further reduction of the marker positioning error. In this application we are considering real-time operations hence, we require real-time predictions. The limited data information and the absence of a suitable predicting model limit our choices for real-time predictions.

### 5.3.1. Extrapolation

A likely real-time methodology for rotor prediction is rotor extrapolation. In order to be able to implement such a system, a more detailed view of the rotor  $R$  is needed. The rotor  $R$  is defined as

$$R = e^B = \cos |B| + \frac{\sin |B|}{|B|} B \quad (5.35)$$

where  $B = B_1(e_2e_3) + B_2(e_3e_1) + B_3(e_1e_2)$ ;  $e_1, e_2, e_3$  are the orthonormal basis vectors and  $|B| = \sqrt{B_1^2 + B_2^2 + B_3^2}$ . Let  $\mathbf{B} = [B_1, B_2, B_3]$ , where  $B_i$  for  $i = 1, 2, 3$  are the bivector coefficients, and  $B' = B/\text{sinc}|B|$  so that  $\mathbf{B}' = [B'_1, B'_2, B'_3]$  where  $B'_i = B_i/\text{sinc}|B|$  are the bivector components. In Geometric Algebra these can be expressed as

$$\begin{aligned} R &= [\alpha \ 0 \ 0 \ 0 \ B_1 \ B_2 \ B_3 \ 0] \\ &= [\cos |B| \ 0 \ 0 \ 0 \ \text{sinc}|B|B'_1 \ \text{sinc}|B|B'_2 \ \text{sinc}|B|B'_3 \ 0] \end{aligned} \quad (5.36)$$

Looking at the structure of the rotor (eq. 5.36), as it is expressed in GA,  $|B| = \alpha \cos(\alpha)$ . The bivector components can then be easily calculated as

$$\mathbf{B}' = \frac{\mathbf{B}}{\text{sinc}|B|} \quad (5.37)$$

More information regarding rotors and Geometric Algebra can be found in [15, 45].

Assume that we want to estimate the rotor at time  $s = 1$ , having the rotors at time steps  $s = 0, 1/3, 2/3$  (see figure 5.4). An extrapolating approach which can calculate an estimate of each time step, according to a reference rotor  $R_0$  (time  $s = 0$ ) is given below.

$$R_s = e^{b_0s^2 + b_1s} R_0 \quad (5.38)$$

hence

$$e^{b_0s^2 + b_1s} = R_s \tilde{R}_0 \quad (5.39)$$

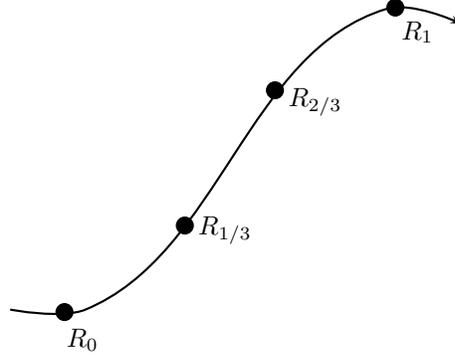


Figure 5.4.: The rotor trajectory over time.

Let  $R_k = R_s \tilde{R}_0$ , where  $k = s$  and  $B'_k$  are the bivector components of the rotor  $R_k$ . Thus,

$$b_0 s^2 + b_1 s = B'_k \quad (5.40)$$

for

$$\begin{aligned} s = 1/3, & \quad \frac{b_0}{9} + \frac{b_1}{3} = B'_{1/3} \\ s = 2/3, & \quad \frac{4b_0}{9} + \frac{2b_1}{3} = B'_{2/3} \\ s = 1, & \quad b_0 + b_1 = B'_1 \end{aligned}$$

It can be easily proved that

$$\begin{aligned} b_0 &= \frac{9B'_{2/3}}{2} - 9B'_{1/3} \\ b_1 &= 6B'_{1/3} - \frac{3B'_{2/3}}{2} \end{aligned}$$

Finally, the estimated current rotor (at time step  $s = 1$ ) can be calculated by rotating the reference rotor  $R_0$  by  $e^{b_0+b_1}$  as given in the equation below.

$$R_1 = e^{b_0+b_1} R_0 \quad (5.41)$$

### 5.3.2. Prediction Filtering

An alternative approach for solving the problem of rotor predictions could be the use of Kalman filtering. Having a more detailed study of the rotor's elements, we observed that in order to estimate the rotor, the trajectory of 4 elements (1 scalar and 3 bivectors) should be tracked. Since small changes in scalar,  $a$ , produce large changes in the resultant rotor, the KF should not be used to track

the scalar (see figure 5.5). On the contrary, the bivector components of the rotor have more constant and smoothed values. Looking at figure 5.6, it is easily seen that the bivector components are more predictable, having similar motion shape-trajectories as the marker positions; hence they can be predicted in the same way. Taking this into consideration, it is reasonable to implement a Kalman filter, similar to the one proposed in section 4.2, which will be able to estimate the future values of the rotor  $R$  over time.

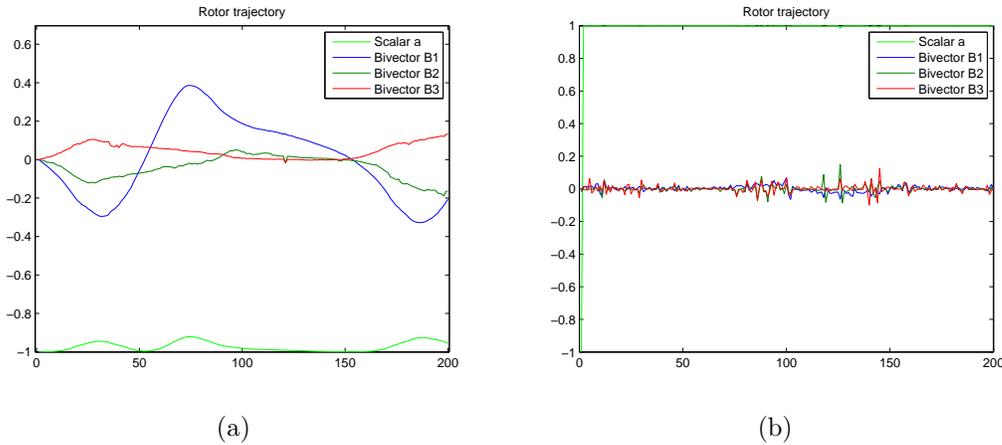


Figure 5.5.: (a) The rotor trajectory and (b) the rotation between two consecutive frames over time.

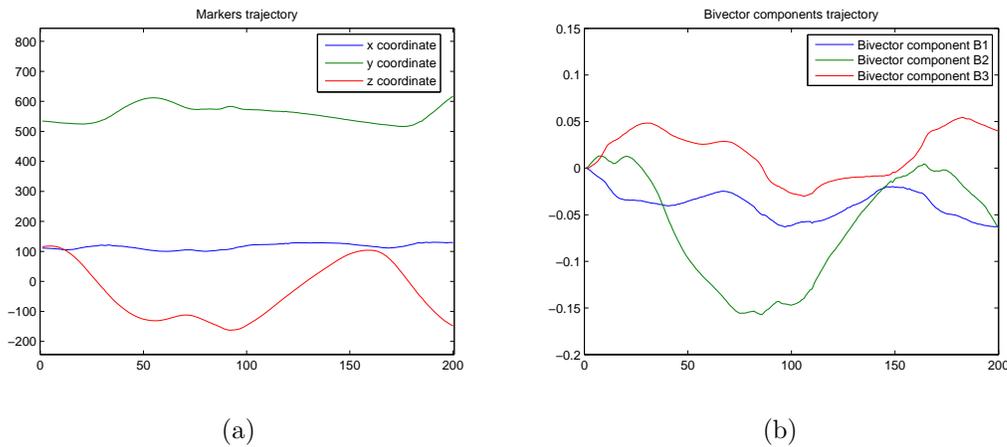


Figure 5.6.: (a) The marker's and (b) the Bivector components' trajectories over time.

First, we have to calculate the bivector components, as shown in equation 5.37, which will be the input state of the KF. The state  $\mathbf{x}_k$  is equal to  $\mathbf{x}_k = [B'_{1,k}, \dot{B}'_{1,k}, \ddot{B}'_{1,k}, B'_{2,k}, \dot{B}'_{2,k}, \ddot{B}'_{2,k}, B'_{3,k}, \dot{B}'_{3,k}, \ddot{B}'_{3,k}]$ , where  $B'_i$  is the value of the bivector component  $i = 1, 2, 3$ ,  $\dot{B}'_i$  is its velocity and  $\ddot{B}'_i$  its acceleration. The

transition function could be a variable turn model (VTM) such as the proposed model in section 5.1.4.

The observation state  $\mathbf{Z}_k$  of the system is the rotor that expresses the rotation between the frame in time  $k - 2$  and  $k - 1$ . It is important at this point to recall that the rotation between the previous and the current frame is re-calculated every time just after the marker prediction using the estimated marker positions. Also, remember that at the end of each prediction, the rotor should be normalised in order to again be a valid<sup>1</sup> rotor.

---

<sup>1</sup>a rotor  $R$  is valid when  $R\tilde{R} = 1$ , and  $\tilde{R}$  is the quaternion conjugate of  $R$

# 6

## Results

---

### 6.1. The Experimental Environment

Experiments were carried out using a 16 camera Phasespace motion capture system capable of capturing data at 480Hz [16]. The algorithms were implemented in MATLAB and run on a Pentium IV PC. The system, proposed in Chapter 4, can process up to 350 *limb segment pairs per second* (using MATLAB). Our datasets comprise both simulated and real data (i.e. captured data with natural occlusions or occlusions generated by artificial deletion) with more than 5000 *frames* in each.

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 and hence lead to erroneous calculation of the model parameters. There are two types of data in our experiments: one consists of 7 segment leg datasets and the other of 5 segment arm datasets. In each case, the motion of the body was studied for a certain number of frames, ensuring rotation occurred at each joint.

### 6.2. Results

In this section, our experimental results will be presented. The results are separated into two sections. This section presents and discusses the results of the

methodology analysed in Chapter 4. Section 6.3 presents, analyses and compares the results between the methodology in Chapter 4 and its filtering extensions in Chapter 5.

Using real data with occlusions generated by artificial deletion, we calculated the error of the proposed methodologies under real conditions; the error is given by the average distance between the true position and the estimated position (marker and the CoR). 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, the more accuracy we achieve. In the following, we will assume that each limb has 3 markers.

### 6.2.1. One missing marker on a limb segment

The most common case is when a single marker is missing; this subsection discusses the results of skeletal reconstruction for the case of one occluded marker. When one marker is missing, information from the current position of neighbouring visible markers is used to estimate the position of the missing marker. Marker positions are estimated with high accuracy, differing from the true position by an average (over 20 runs) of  $2.6mm$ . The CoR is calculated using three different approaches; the first calculates the CoR location using the predicted marker positions, the second calculates the CoR using only information about the limb segment on which all markers are visible with  $\bar{\mathbf{a}}_w$  updated using the predicted data and the third calculates the CoR using information from the limb segment on which all markers are visible but uses the last available  $\bar{\mathbf{a}}_w$  ( $\bar{\mathbf{a}}_w$  is not updated and it is assumed that it remains constant over time; see section 3.4 for more details). Table 6.1 lists the average results for the case where one marker on a limb segment is missing. Obviously, the best results are obtained where the CoR location is calculated using the information from the limb segment with all its markers visible. Thus, the CoR is updated from information from the limb with all markers visible but we note that  $\bar{\mathbf{a}}_w$  is updated using the predicted positions of the markers.

### 6.2.2. Two missing markers on a limb segment

When two markers on a limb segment are missing then information from the remaining visible neighbouring marker is used. We take advantage of the fact that markers have approximately constant inter-marker distance and apply constraints to ensure that this fact will always be satisfied. The average distance between the true and the estimated marker positions for the case where two markers are missing is approximately  $6.5mm$ ; while the difference in the CoR location between the true and the estimated position is less than  $1cm$ . Table 6.2 shows the average results for this case. Again, the lowest error is seen when the CoR location is

Table 6.1.: Average results (over 20 runs) on real data with occlusions generated by deletions. Case of one missing marker on a limb segment for more than 1500 frames.

The error when one missing marker is entirely occluded	
	Error ( <i>mm</i> )
Marker position	2.584124
CoR (using the predicted marker positions)	7.044406
CoR (when $\bar{\mathbf{a}}_w$ is updated using the predicted data.)	5.905546
CoR (when $\bar{\mathbf{a}}_w$ is not updated using the predicted data.)	6.742417

calculated using data from the limb segment which has all its markers visible and  $\bar{\mathbf{a}}_w$  is updated using the predicted marker positions.

Table 6.2.: Average results (over 10 runs) on real data with occlusions generated by deletions. Case of two missing markers on a limb segment for more than 1500 frames.

The error when 2 missing markers are entirely occluded	
	Error ( <i>mm</i> )
Marker position	6.549751
CoR (using the predicted markers positions)	12.587746
CoR (when $\bar{\mathbf{a}}_w$ is updated using the predicted data.)	9.544872
CoR (when $\bar{\mathbf{a}}_w$ is not updated using the predicted data.)	10.809371

### 6.2.3. All markers on a limb segment are missing

When all markers on a limb segment are occluded, the marker positions and the CoR can be estimated using two different but related approaches utilising the available marker locations on the other limb segment. If some markers on the other limb segment are visible, the missing marker positions can be calculated using the CoR estimate,  $\hat{\mathbf{C}}_k$ , as proposed in section 4.3.4. The error between the true and estimated marker positions for this case is less than  $2cm$ , whereas the difference between the true and estimated CoR location is approximately  $1.2cm$ . These results are presented in table 6.3. It is also observed that when the predicted marker positions are used to update  $\bar{\mathbf{a}}_w$ , the noise added to the system is more than that introduced by just using the last available position values,

thereby increasing the error.

On the other hand, if both limb segments have all their markers occluded, only information from previous frames may be used. Hence, if markers are occluded for extended time periods during which rapid changes in direction or/and velocity occur, the predictor fails to provide a useful estimate of the position of missing markers or the reconstructed CoR location. The average error for that case is more than  $10cm$  for marker positions and almost  $3cm$  for the CoR.

Table 6.3.: Average results (over 20 runs) on real data with occlusions generated by deletions. Case where all markers on a limb segment are missing for more than 1500 frames and some markers on the other limb segment are visible.

The error when 3 missing markers are entirely occluded	
	Error ( <i>mm</i> )
Marker position	19.452111
CoR (when $\bar{\mathbf{a}}_w$ is updated using the predicted data.)	17.854712
CoR (when $\bar{\mathbf{a}}_w$ is not updated using the predicted data.)	12.105377

Looking carefully at the results, it is clear that the Kalman filter gives reasonable estimates except where all markers on a limb are occluded. In that case, the system fails because the observation vector is calculated using approaches based only on previous frames. In the first approach the algorithm assumes that the velocity of the occluded markers remains stable over time, whereas in the second approach it assumes that the rotation between two neighbouring frames will not be changed. Figure 6.1 shows how the marker position errors using the Kalman filter for both constant velocity and constant rotation models compare for different numbers of missing markers. It is obvious that the error increases dramatically in the case where 3 markers on the same limb are non-visible; however, when the constant rotation approach is used, the estimated positions of the missing markers have less error from the true position in comparison with using a constant velocity model. This is because use of the constant rotor approach will ensure all markers have been rotated using the same rotor, thus, markers will not follow inconsistent trajectories. Alternatively, when a constant velocity model is used, each marker can have its own trajectory “violating” the skeletal model of the tracked limb.

#### 6.2.4. Markers visible by one camera

Looking at numerous real datasets, we have observed a high probability that a missing marker is visible in one camera. Most of the time, when a marker is not located by the MoCap system it is not entirely missing, information about its

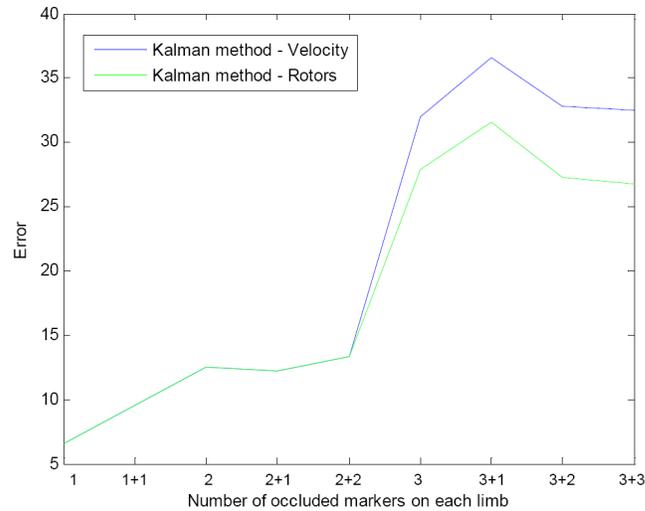


Figure 6.1.: The Kalman filter error using the *constant velocity* and the *rotors* approaches, against the number of occluded markers on each limb segment. For example, 2 + 1 means 2 occluded markers on the first limb segment and 1 occluded marker on the second.

position is often returned by a single camera. This information identifies a line starting from the camera and passing through the position of the occluded marker. Hence, by relaxing the constraints that the inter-marker distances are constant and accepting that the real position of the marker is on the line returned by the camera, we might obtain a more reliable position estimate. Using the model proposed in section 4.3.5, we observe that the results are much more accurate than for the case where this information has been ignored. Table 6.4 lists the average results (over 10 runs) in the case of one missing marker on each limb segment. Table 6.5 presents the results when only one marker is fully visible and two markers are partially visible in one camera, while table 6.6 shows the results when one marker is fully visible, one marker is partially visible in one camera and one marker is entirely missing. The error is reduced by more than 70% in the first case and almost 40% in the second case compared with the corresponding cases when this information is not used. Another important observation is that, 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. The reason for this is obvious. If we first predict the position of the entirely missing marker, the position of the partially visible marker will be calculated having introduced another noise element compared to the reverse case. This happens because our algorithm predicts a missing marker more accurately when some information about its position is available.

Figures 6.2 show examples of the output of the proposed algorithms on real data; it is the lower body of a female as she runs. These illustrative examples

Table 6.4.: Average results (over 10 runs) on real data with occlusions generated by deletions. Case of one missing marker on each limb segment for more than 1500 frames.

The error when missing markers are visible by one camera	
	Error ( <i>mm</i> )
Marker position	0.585351
CoR (using the predicted markers positions)	1.281958

Table 6.5.: Average results (over 10 runs) on real data with occlusions generated by deletion. Case of one visible and two partially visible markers on one limb segment. Both markers are missing for more than 1500 frames.

The error when missing markers are visible by one camera	
	Error ( <i>mm</i> )
Marker position	1.973166
CoR (using the predicted markers positions)	2.571261

show that the proposed method can estimate the position of the occluded markers and locate the joint parameters in real-time with high accuracy.

Figure 6.3 shows an example of the true and predicted positions of an occluded marker and the CoR for the case of a single occlusion and the case of a missing marker visible by one camera, respectively. Its clear that the occluded marker can be tracked with high accuracy when it is visible in a camera and its CoR position can be reconstructed efficiently even if the occlusion period exceeds 1500 frames. Figure 6.4 also shows an example of the error variation over time due to occlusion for the case of 2 missing markers. The error after the occlusion is not set to zero because the rotor  $R^{1,k}$  expressing the rotation between the first and the current frame, is an estimation using the predicted markers positions. Hence, even if all the markers are now available, this rotor will still be an estimation. The position of the CoR is based on that rotor (see equation 3.33), and hence is also an estimation of the true position. However, the error of the CoR is reduced during time (when all markers are visible) because the new positions of the markers give a more accurate estimation of the rotor.

Table 6.6.: Average results (over 10 runs) on real data with occlusions generated by deletion. Case of one entirely missing marker, one partially visible marker and one visible marker on one limb segment. Both markers are occluded for more than 1500 frames.

The error when missing markers are visible by one camera	
	Error ( <i>mm</i> )
Marker position	3.498871
CoR (using the predicted markers positions)	8.369646

### 6.3. The filtering extended methodologies

In this section we compare and discuss the results of the methodology presented in Chapter 4 and its filtering extension in Chapter 5. The first step of the filtering extension was to find a more reliable dynamic model for the Kalman filtering. Therefore, we implemented a variable turn model (VTM) which uses the velocity and acceleration of the  $x, y, z$  components of each individual marker. Using this sophisticated dynamic model, we took into consideration any changes in the velocity, direction or acceleration of the markers' trajectories. Using a VTM model, we reduced the error between the real and the estimated position of the markers by 1.3% on average when 1 marker is missing compared to the simple constant velocity (CV) model. However, we also observe a 12% increase in processing time. This small time addition does not affect our target for real-time implementation because we can still process approximately 305 *limb segment pairs per second*. On the other hand, when 2 or 3 markers are missing there is an average error increase of 15% and 5% respectively. At the moment it is unclear why VTM model increase the error; it seems that, the most reasonable explanation is that, if two or more markers on a limb segment are not visible, the additional noise introduced to the system is bigger. These results are presented in table 6.7.

A further reduction of the error was achieved when the unscented transformation was implemented through Kalman filtering. Such a model is necessary to process nonlinear systems, like those studied here. The UKF reduced the error by 7.9% when 1 marker is missing, 1.71% when 2 markers are missing and 0.23% when all markers are missing, providing more accurate marker position estimates. An important observation of using a UKF in such systems is that the processing time could be a drawback of this methodology, in this case increasing the processing time by 20% (see table 6.9). However, even under these conditions it is possible to achieve the target of real-time application. Table 6.8 lists the aver-

Table 6.7.: The average error and its % reduction using a VTM instead of a CV model.

Comparison of the results before and after the VTM filtering				
	Marker's Error ( <i>mm</i> )	Change	CoR's* Error ( <i>mm</i> )	Change
One marker <sup>†</sup>	2.5464	-1.3%	6.9214	-1.75%
Two markers <sup>†</sup>	7.5129	+15.0%	15.1052	+20.0%
All markers <sup>†</sup>	20.4520	+5.14%	18.0335	+1.00%

\* The CoR is calculated using the predicted marker positions.

† Missing markers are entirely occluded.

age error and the % reduction using the proposed UKF (with a VTM transition function) instead of the KF (with a constant velocity model) for the case of one missing marker, two missing markers and all the markers occluded.

Table 6.8.: The average error and its % reduction using a UKF (with VTM) instead of a KF (with CV).

Comparison of the results before and after the UKF filtering				
	Marker's Error ( <i>mm</i> )	Change	CoR's* Error ( <i>mm</i> )	Change
One marker <sup>†</sup>	2.3741	-7.90%	6.0440	-14.20%
Two markers <sup>†</sup>	6.4377	-1.71%	12.3120	-2.19%
All markers <sup>†</sup>	19.4012	-0.23%	17.7752	-0.59%

\* The CoR is calculated using the predicted marker positions.

† Missing markers are entirely occluded.

Having a more detailed look at the results, we observe that the UKF performs better in cases where the estimation error of the KF is big. The estimation error is equal to  $\mathbf{e} = \sqrt{(\mathbf{x} - \hat{\mathbf{x}})^2 + (\mathbf{y} - \hat{\mathbf{y}})^2 + (\mathbf{z} - \hat{\mathbf{z}})^2}$ , where  $\mathbf{e}$  is the error,  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$  are the coordinates of the true positions of the marker and  $\hat{\mathbf{x}}$ ,  $\hat{\mathbf{y}}$ ,  $\hat{\mathbf{z}}$  are the coordinates of the estimated positions. For example, when the positioning error of the marker using the KF model is less than  $3.5mm$ , the reduction achieved using the UKF approach is approximately 1.5%. On the contrary, when the error is greater than  $3.5mm$  from the true position, the UKF performs better, achieving an error reduction of 10%. Figure 6.5 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 the estimation error is less than or equal to  $3mm$  is approximately 0.47 for the KF and VTM, while it is 0.68 for the case of the UKF. From figure 6.5, we see that the median estimation error of the markers for the cases of the KF and VTM is approximately 3.17 (6.48 for the CoR), where the corresponding median error for the case of the UKF is approximately 2.35 (5.05 for the CoR).

The last proposed extension was the rotor prediction method. A rotor expresses the rotation between a reference frame and the current frame of a limb segment. In section 4.3 we assumed that the rotation between two consecutive frames remains constant. In order to overcome this assumption, we proposed two possible methods of rotor prediction. However, both extrapolation and filtering prediction of the rotors do not offer any error reduction. Even though the rotor prediction works reasonably using the Kalman theory, the 50% addition of processing time remains a big drawback. Since that the rotor is re-calculated in each frame using the predicted positions of the markers, any rotor estimation will result in high processing time.

Table 6.9 presents the average results of each extended algorithm with the required processing time compared to the proposed KF methodology.

Table 6.9.: Average results after implementing the extension over the proposed methodology.

Comparison of the results before and after the filtering extension. Case of 1 missing marker on a limb segment.			
	Markers Error	CoR Error	Proc. time
Variable Turn Model (VTM)	-1.31%	-1.02%	+12%
Unscented kalman Filter (UKF)	-7.90%	-14.20%	+20%
Rotor Prediction - Extrapolation	+0.15%	+0.094%	+3%
Rotor Prediction - Kalman filtering	-0.075%	-0.05%	+50%

## 6.4. Discussion

The work outlined in this Chapter investigates reconstruction of marker position for multiple cases of occlusion. The 3D location of the markers can be reliably reconstructed even when a single marker occlusion exists for more than 1000 *frames* at a time, returning mean position errors of less than  $3.5mm$ . The position of the CoR using the predicted marker positions can be calculated with a mean error of approximately  $6.35mm$  in cases where one marker on each limb segment is entirely occluded, this increases to  $11.8mm$  in cases where 2 out of 3 markers on a limb are not visible. However, in the case where one of the limb segments

has all its markers available, the CoR can be calculated with higher accuracy using information only from that limb segment, as in equation 3.35, where  $\bar{\mathbf{a}}_w$  is now updated using the predicted positions of the markers. The error between the true CoR location and the estimate for that instance is  $5.9mm$  when one marker is occluded and  $9.5mm$  when two markers are missing. This error is significantly decreased to  $0.6mm$  for marker position estimates and  $1.3mm$  for the CoR position estimates in the case where the missing markers are visible by one camera. These results can only be compared with post-processing methods. Most of the previously proposed missing marker replacement algorithms used within motion capture systems are post-processing algorithms, thus limiting their application. In this work, the methods proposed are for real-time applications ensuring continuous data-flow.

The main advantages of the methods proposed are that the marker positions can be calculated in real-time using an integrated framework (Kalman filter) in combination with inferred information from the neighbouring visible markers. Taking advantage of the fact that the inter-marker distances are approximately constant and where possible, visibility of markers in a single camera, an efficient method for estimating the missing markers and reconstructing the CoR is presented. However, one drawback of this work is the inevitable failure of the missing marker position predictions when all markers are non-visible, thereby leading to erroneous estimation of the CoR.

Certain drawbacks of the proposed methodology can be decreased using more evolved and sophisticated methods during the Kalman Filter implementation. The use of a variable turn model (VTM), instead of a simple constant velocity model, can give significant improvements in cases where the trajectories of the markers are variable and have abrupt fluctuations in speed and direction. A parallel use of the unscented transformation and the VTM, as proven in our experiments, can reduce the error by 7.90% in the case of one missing marker, 1.71% when 2 markers are missing and 0.23% when all markers are occluded. This also effects the estimation of the CoR with a 14.2% decrease in the case of one marker, 2.19% decrease in the case of two markers and 0.59% decrease in the case where all markers are not visible to the cameras.

The use of a rotor predictor, instead of assuming that the rotation between two consecutive frames remains constant, does not decrease the error as might have been expected. On the contrary, the processing time of such filtering is high and jeopardizes the target of real-time application.

Experiments demonstrate that the proposed method and its extensions effectively recovers, in real-time, good estimates of the true positions of the missing markers and reconstructs the CoR locations under multiple occlusions even if markers are not visible for extended periods of time. Some of these results have been published in [56], and [57].

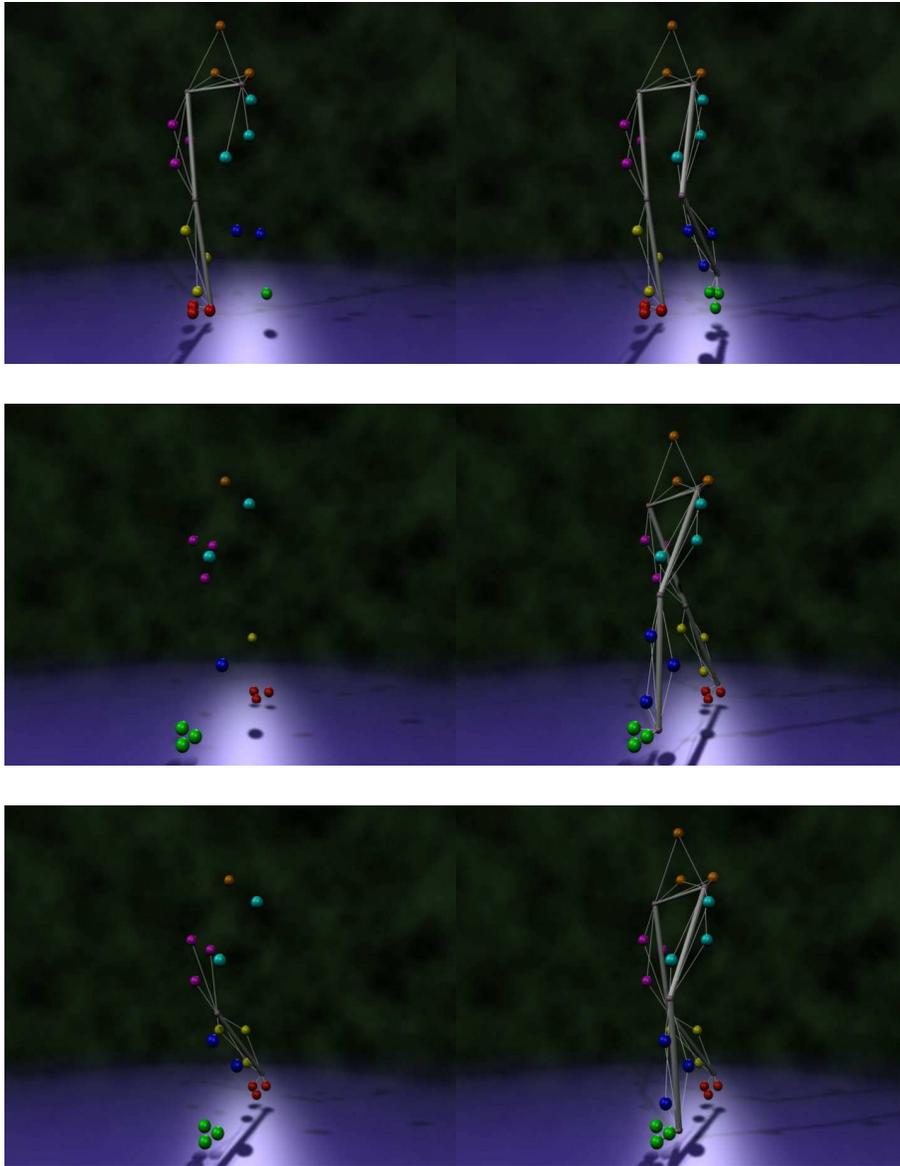


Figure 6.2.: Examples of implementation on real data (Lower body). On the left side of the picture there are some markers which are missing; on the right side of the image these markers have been correctly estimated.

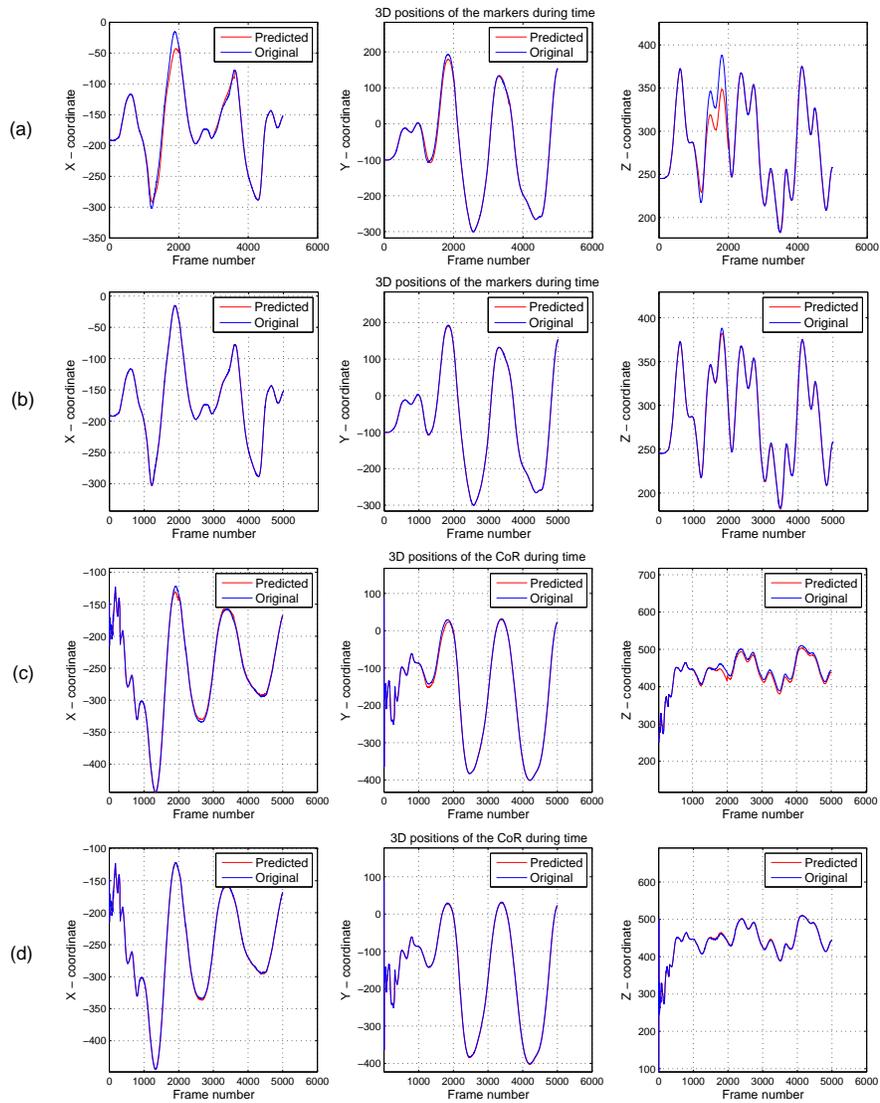


Figure 6.3.: An example of the predicted 3D positions and the true coordinates of the *Markers* in the case (a) of one missing marker on a limb segment and (b) a missing marker visible by one camera. (c) and (d) show the predicted 3D positions of the *CoR* under the same conditions. The occlusion periods are between frames 1000-2000 and 3000-3600.

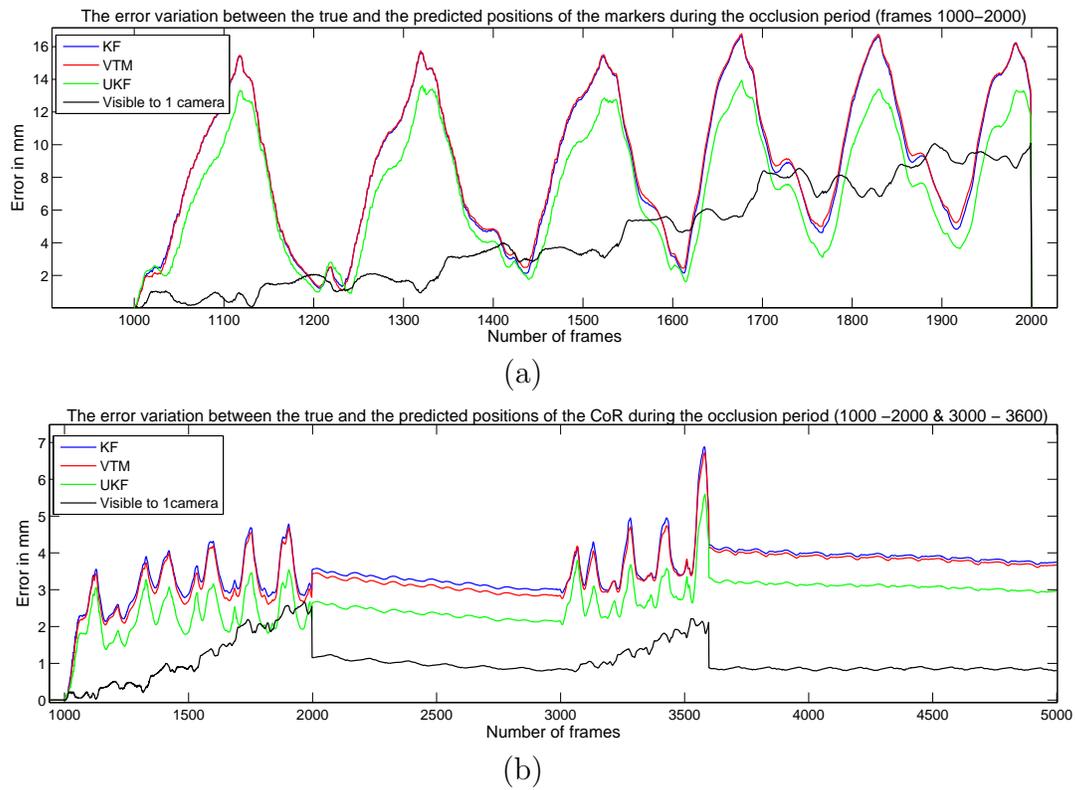


Figure 6.4.: The error between the predicted and the true positions of the (a) *Markers* and (b) *CoR* respectively during the occlusion period.

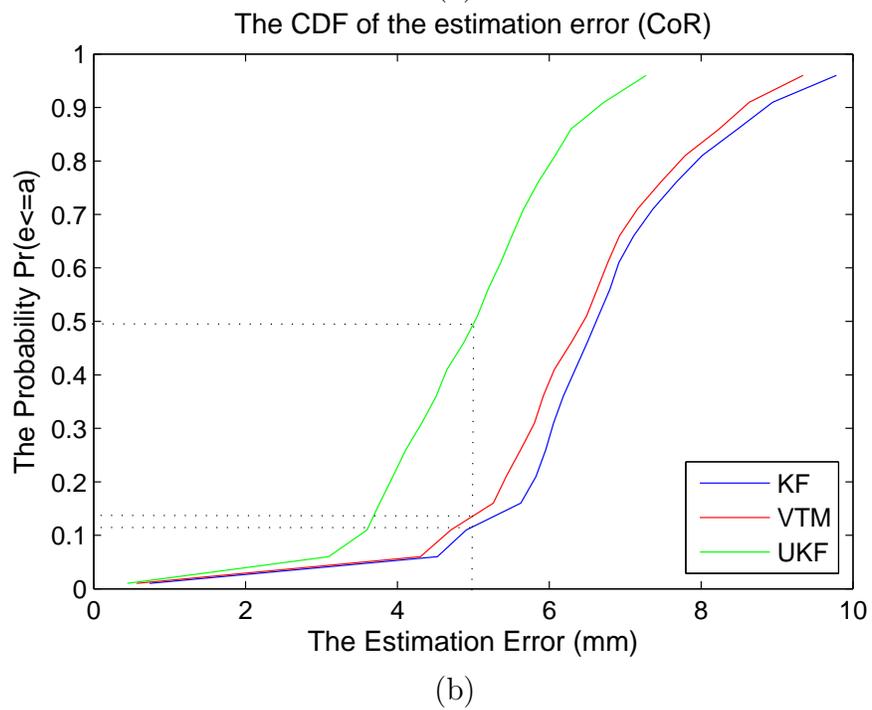
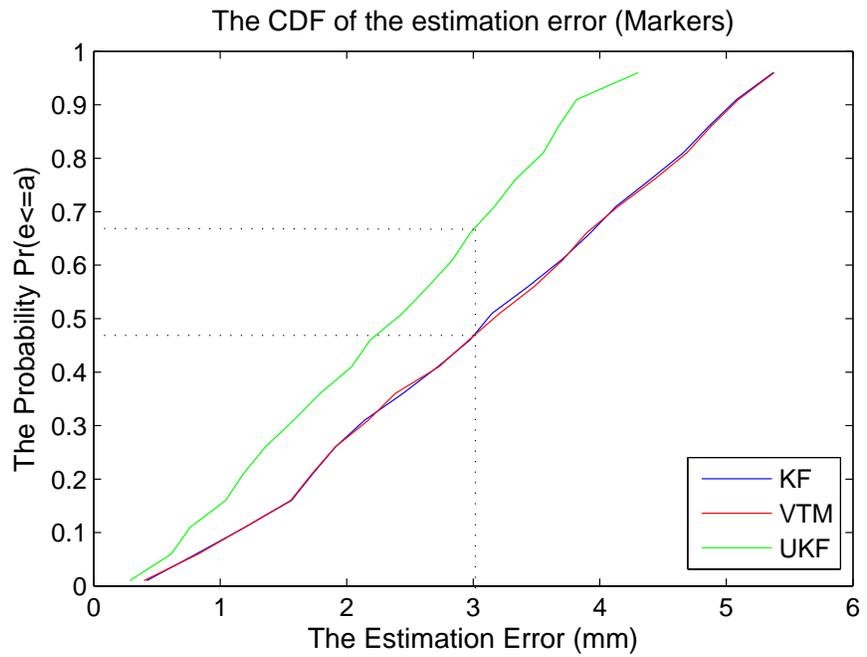


Figure 6.5.: The CDF of the estimation error (a) of the markers and (b) of the CoR for the case of one missing marker on each limb segment.

# 7

## Conclusions and Future Work

---

This report concerns algorithms related to the problem of using marker-based optical motion capture data to automatically establish a skeleton model to which the markers are attached. It is divided into three sections; the first section describes the mathematical and experimental framework including the cameras used for the experiments, the association between the markers and the limbs, and the clustering of marker data into groups corresponding to the limb segments to which they are attached. Section 2 considers the problem of fitting skeletal models to marker-based optical motion capture data. Using the well known *Procrustes* formulation, it shows how to establish estimates of the relative orientation of a limb in each frame relative to a reference frame. Thereafter, it estimates the joint location and identifies the optimal skeleton in real-time. The implemented method takes advantage of the approximation that all markers on a segment are attached to a rigid body, acquiring the skeleton model from a stream of motion capture data. This allows a closed form sequential algorithm for estimating the centre of rotation of a skeletal model in real time. However, this algorithm neglects frames containing missing markers.

A common phenomenon during motion capture, even with many cameras, is marker occlusion by elements of the scene, leading to missing data. In order to unambiguously establish the marker position, each marker must be visible to at least two cameras in each frame. The third section of this report presents a prediction method which copes with such instances. It proposes a real-time approach for estimating the position of occluded markers using previous positions and information inferred from an approximate rigid body assumption. With a

continuous stream of accurate 3D data, we can perform real-time CoR estimation, thereby producing skeletal information for use in visual performance feedback. 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. Also, the proposed system uses the information returned by each single camera regarding the position of missing markers. Experiments demonstrate that the method presented effectively recovers good estimates of the true positions of the missing markers, even if large sequences with occluded data exist, in which more than 1 marker is occluded on each limb, and also when the limb rapidly changes direction. The system developed is suitable for real time processing and provides excellent estimates of the markers' position and the CoR location.

Chapter 5 presents a filtering extension of the proposed methodology. Using a more sophisticated dynamic model (VTM) which takes account of the velocity and the acceleration of the coordinates of each individual marker, we reduced the error by 1.3% with a time processing addition of 12%. A further reduction can be achieved using the Unscented Transformation in cooperation with a Kalman filter. The UKF reduces the marker's error by 7.9% and the CoR's error by 14.2 when one marker is missing, however increases the processing time by 20%. On the contrary, both extrapolation and filtering prediction of the rotors, which express the rotation between a reference frame and the current frame of a limb segment, do not offer the expected error decrease. Even if the rotor prediction seems to work reasonably using the Kalman theory, the 50% addition of processing time remains a serious drawback. Since the rotor is re-calculated in each frame using the predicted positions of the markers, any rotor estimation will have high processing time.

Future work will introduce biomechanical constraints to restrict motions to those from a feasible set. Hence, the Kalman filter should be extended by utilising a more sophisticated analysis of human motion. The use of accelerometers and inertial sensors will also be studied. These sensors will be able to provide additional information related to the limb orientation and the markers' acceleration, creating a more reliable system.

# APPENDICES





# A

## Alternative Cost Function

---

Appendix A presents an alternative cost function used for calculating the CoR. This new approach attempts to incorporate a model of noise with the cost function used. For the simple derivation presented here, noise is assumed to be isotropic Gaussian, although the form of the derivation also allows for non-isotropic cases.

It is assumed that 3 markers are available on either side of the joint. The CoR in a given frame may be expressed as (mean of estimates from both sides)

$$C^k = \frac{1}{6} \left[ \sum_{i=1}^3 \left( x_i^k - R_x^k a_x^i \tilde{R}_x^k \right) + \sum_{j=1}^3 \left( y_j^k - R_y^k a_y^j \tilde{R}_y^k \right) \right] \quad (\text{A.1})$$

If we assume that the true rotors, vectors and location of the CoR are available then the error may be expressed as

$$S_{new}^k = \sum_{l=1}^3 \left( C^k + R_x^k a_x^l \tilde{R}_x^k - x_l^k \right)^2 + \sum_{m=1}^3 \left( C^k + R_y^k a_y^m \tilde{R}_y^k - y_m^k \right)^2 \quad (\text{A.2})$$

This corresponds to measuring the sum squared error between the measured marker locations and their locations assuming all parameters are known.

Substituting (A.1) into (A.2) and differentiating w.r.t. the vectors gives

$$\begin{aligned} \partial_{a_x^l} S_{new}^k &= 2a_x^l - \bar{a}_x - \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k - 2\tilde{R}_x^k x_l^k R_x^k + \tilde{R}_x^k \bar{x}^k R_x^k + \tilde{R}_x^k \bar{y}^k R_x^k \\ &= 0 \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned} \partial_{a_y^l} S_{new}^k &= 2a_y^l - \bar{a}_y - \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k - 2\tilde{R}_y^k y_l^k R_y^k + \tilde{R}_y^k \bar{y}^k R_y^k + \tilde{R}_y^k \bar{x}^k R_y^k \\ &= 0 \end{aligned} \quad (\text{A.4})$$

Now, the equivalent result using the conventional transformational algorithm (as presented in Chapter 3) may be written

$$\begin{aligned}\partial_{a_x^l} S^k &= -\tilde{R}_x^k x_l^k R_x^k + \tilde{R}_x^k \bar{y}^k R_x^k + a_x^l - \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k \\ &= 0\end{aligned}\tag{A.5}$$

The two resulting systems of linear equations may be easily shown to be equivalent. For example, taking vector  $a_x^1$

$$\partial_{a_x^1} S^k = \frac{1}{2} \left[ \frac{1}{3} (\partial_{a_x^1} S_{new}^k + \partial_{a_x^2} S_{new}^k + \partial_{a_x^3} S_{new}^k) + \partial_{a_x^1} S_{new}^k \right]\tag{A.6}$$

Thus from the point of view of estimating the vectors (stage 1 of a given skeleton fitting algorithm) this new cost function has the same optimum as that for the conventional transformational algorithm.

It is interesting to note that when considering using the cost functions to refine the rotors within an iterative optimisation algorithm, the differential w.r.t. the rotors' bivector parameters are not equivalent.

$$\partial_{B_x} S^k = -2 \sum_{i=1}^3 \sum_{j=1}^3 \partial_{B_x} \left[ \left( R_x^k a_x^i \tilde{R}_x^k \right) \cdot b_{ij}^k + \left( R_x^k a_x^i \tilde{R}_x^k \right) \cdot \left( R_y^k a_y^j \tilde{R}_y^k \right) \right]$$

and now,

$$\begin{aligned}\partial_{B_x} S_{new}^k &= \sum_{i=1}^3 \left\{ -\partial_{B_x} \left[ \left( R_x^k \left( a_x^i - \frac{1}{2} \bar{a}_x \right) \tilde{R}_x^k \right) \cdot \left( \frac{1}{2} \bar{x}^k + \frac{1}{2} \bar{y} - x_i^k - \frac{1}{2} R_y^k \bar{a}_y \tilde{R}_y^k \right) \right] \right. \\ &\quad \left. + \partial_{B_x} \left[ \left( R_x^k \left( \frac{1}{2} \bar{a}_x \right) \tilde{R}_x^k \right) \cdot \left( \frac{1}{2} \bar{x}^k + \frac{1}{2} \bar{y}^k - y_i^k - R_y^k \left( \frac{1}{2} \bar{a}_y - a_y^i \right) \tilde{R}_y^k \right) \right] \right\}\end{aligned}$$

where  $R_x^k = \exp(B_x^k)$  and  $B_x^k$  is the bivector associated with the  $x$ -limb rotation in frame  $k$ .

In conclusion, we have shown that this alternative approach based on the assumption of isotropic Gaussian noise and a perfect model gives identical results to the simple model described in Chapter 3. This is true when the rotors are estimated independent of the assumption of a skeleton model. It is also interesting to note that if the rotors are estimated jointly with the skeleton parameters, the 2 cost functions are not equivalent. This should be investigated further although such an optimum may not be found in real-time situations and so does not lie within the scope of this report.

# List of Abbreviations

---

CoR	-	Centre of Rotation
CV	-	Constant Velocity
CDF	-	Cumulative Distribution Function
DOF	-	Degrees of Freedom
EKF	-	Extended Kalman Filter
GA	-	Geometric Algebra
GPLVN	-	Gaussian Process Latent Variable Model
KF	-	Kalman Filter
MMSE	-	Minimum Means Square Error
MoCap	-	Motion Capture
NCT	-	Nearly Constant Turn
OCGA	-	Oriented Conformal Geometric Algebra
RMMSE	-	Recursive Minimum Means Square Error
RV	-	Random Variable
SUT	-	Scaled Unscented Transform
UKF	-	Unscented Kalman Filter
UT	-	Unscented Transform
VTM	-	Variable Turn Model



# Bibliography

---

- [1] Junichi Hashiguchi, Hiroki Nivomiya, Hiroshi Tanaka, Mari Nakamura, and Katsuya Nobuhara. Biomechanical analysis of a golf swing using motion capture system. In *Proceedings of Annual Meeting of Japanese Society for Orthopaedic Biomechanics*, volume 27, pages 325–330, 2006.
- [2] Motion Reality golf systems, <http://www.motionrealitygolf.com>.
- [3] Maureen K. Holden, Thomas A. Dyar, Lee Schwamm, and Emilio Bizzi. Virtual-environment-based telerehabilitation in patients with stroke. *Presence: Teleoper. Virtual Environ.*, 14(2):214–233, 2005.
- [4] Lamberto Piron, Paolo Tonin, Francesco Piccione, Vincenzo Iaia, Elena Trivello, and Mauro Dam. Virtual environment training therapy for arm motor rehabilitation. *Teleoperators and Virtual Environments*, 14(6):732–740, 2005.
- [5] Jurgen Broeren, Katharina S. Sunnerhagen, and Martin Rydmark. A kinematic analysis of a haptic handheld stylus in a virtual environment: a study in healthy subjects. *Journal of NeuroEngineering and Rehabilitation*, 4:13, May 2007.
- [6] Alberto Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [7] Marius-Calin Silaghi, Ralf Plänkers, Ronan Boulic, Pascal Fua, and Daniel Thalmann. Local and global skeleton fitting techniques for optical motion capture. In *CAPTECH '98: Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, pages 26–40, London, UK, 1998. LNCS, Springer Verlag Heidelberg.
- [8] K. Halvorsen, M. Lesser, and A. Lundberg. A new method for estimating the axis of rotation and the center of rotation. *Journal of Biomechanics*, 32:1221–1227, 1999.
- [9] Andrea Cereatti, Valentina Camomilla, and Aurelio Cappozzo. Estimation of the centre of rotation: a methodological contribution. *Journal of Biomechanics*, 37:413–416, March 2004.
- [10] K. Halvorsen. Bias compensated least squares estimate of the center of rotation. *Journal of Biomechanics*, 36:999–1008(10), July 2003.

- 
- [11] S. Holzreiter. Calculation of the instantaneous centre of rotation for a rigid body. *Journal of Biomechanics*, 24(7):643–647, 1991.
- [12] James F. O’Brien, Robert E. Bodenheimer, Gabriel J. Brostow, and Jessica K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proceedings of Graphic Interface*, pages 53–60, 2000.
- [13] Rainald M. Ehrig, William R. Taylor, Georg N. Duda, and Markus O. Heller. A survey of formal methods for determining the centre of rotation of ball joints. *Journal of Biomechanics*, 39(15):2798–2809, 2006.
- [14] Jonathan Cameron and Joan Lasenby. A real-time sequential algorithm for human joint localization. In *ACM SIGGRAPH Posters*, page 107, New York, USA, 2005. ACM Press.
- [15] Anthony N. Lasenby, Joan Lasenby, and Richard Wareham. A covariant approach to geometry using geometric algebra. Technical Report F-INFENG/TR-483, Cambridge University Engineering Department, 2004.
- [16] PhaseSpace Inc.: Optical motion capture systems, <http://www.phasespace.com>.
- [17] CodaMotion. Optical motion capture systems, <http://www.codamotion.com>.
- [18] Vicon Motion System and Peak Performance. Motion capture systems, <http://www.vicon.com>.
- [19] The Mathworks MATLAB and Simulink for Technical Computing, <http://www.mathworks.com>.
- [20] Leo Dorst. First order error propagation of the Procrustes method for 3d attitude estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):221–229, February 2005.
- [21] Sahan S. Hiniduma Udugama Gamage and Joan Lasenby. New least squares solutions for estimating the average centre of rotation and the axis of rotation. *Journal of Biomechanics*, 35(1):87–93, January 2002.
- [22] P. Cerveri, N. Lopomo, A. Pedotti, and G. Ferrigno. Derivation of centers and axes of rotation for wrist and fingers in a hand kinematic model: Methods and reliability results. *Annals of Biomedical Engineering*, 33(3):402–412, January 2005.
- [23] P. Cerveri, E. De Momi, N. Lopomo, G. Baud-Bovy, R. M. L. Barros, and G. Ferrigno. Finger kinematic modeling and real-time hand motion estimation. *Annals of Biomedical Engineering*, 35(11):1989–2002, November 2007.
- [24] Adam G. Kirk, James F. O’Brien, and David A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 782–788, June 2005.
- [25] Berthold Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, April 1987.

- [26] Chris Doran and Anthony Lasenby. *Geometric Algebra for Physicists*. Cambridge University Press, Cambridge UK, 2003.
- [27] Sahan S. Hiniduma Udugama Gamage and Joan Lasenby. A new least squares solution for estimation of centre and axis of rotation. Technical Report F-INFENG/TR. 399, CUED, 2001.
- [28] Maurice Ringer and Joan Lasenby. A procedure for automatically estimating model parameters in optical motion capture. In *Proceedings of the British Machine Vision Conference*, pages 747–756, 2002.
- [29] Douglas J. Wiley and James K. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications*, 17(6):39–45, 1997.
- [30] Charles Rose, Michael Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
- [31] Jean-Christophe Nebel. Keyframe animation of articulated figures using autocollision-free interpolation. In *Proceedings of the 17th Eurographics UK Conference'99*, 13-15 April 1999.
- [32] Golam Ashraf and Kok Cheong Wong. Dynamic time warp based framespace interpolation for motion editing. In Sidney Fels and Pierre Poulin, editors, *Graphics Interface*, pages 45–52. Canadian Human-Computer Communications Society, 2000.
- [33] Jean-Christophe Nebel. Keyframe interpolation with self-collision avoidance. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation*, pages 77–86. Springer, September 1999.
- [34] Arjen Van Rhijn and Jurriaan D. Mulder. Optical tracking and automatic model estimation of composite interaction devices. *IEEE Virtual Reality Conference (VR'06)*, 00:135–142, 2006.
- [35] Klaus Dorfmüller-Ulhaas. Robust optical user motion tracking using a Kalman filter. Technical Report TR-2003-6, Institut fuer Informatik, Universitätsstr. 2, 86159 Augsburg, May 2003.
- [36] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci. The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments. In *Virtual Reality Software and Technology, VRST, ACM*, pages 1–10, December 20-22 1999.
- [37] Lorna Herda, Pascal Fua, Ralf Plänklers, Ronan Boulic, and Daniel Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proceedings of the IEEE Computer Animation (CA'00)*, pages 77–86, Philadelphia, Pennsylvania, USA, May 3-5 2000. IEEE Computer Society.

- [38] Lorna Herda, Pascal Fua, Ralf Plänkers, Ronan Boulic, and Daniel Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science Journal*, 20(3):313–341, 2001.
- [39] Alexander Hornung and Sandip Sar-Dessai. Self-calibrating optical motion tracking for articulated bodies. In *Proceedings of the IEEE Conference on Virtual Reality, VR '05*, pages 75–82, Washington, DC, USA, 2005. IEEE Computer Society.
- [40] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. In *SIGGRAPH '04: ACM Transactions on Graphics*, pages 522–531, New York, NY, USA, August 2004. ACM.
- [41] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. *Proceedings of ACM SIGGRAPH'05: Transactions on Graphics*, 24(3):686–696, 2005.
- [42] Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan. Human motion estimation from a reduced marker set. In *I3D '06: Proceedings of the Symposium on Interactive 3D graphics and games*, pages 35–42, New York, NY, USA, 2006. ACM.
- [43] Guodong Liu and Leonard McMillan. Estimation of missing markers in human motion capture. *The Visual Computer*, 22(9-11):721–728, September 2006.
- [44] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME - Journal of Basic Engineering*, pages 35–45, 1960.
- [45] Andreas Aristidou. Real-Time Estimation of Missing Markers in Optical Motion Capture Data, 1st year report, Cambridge University Engineering Department. Cambridge, UK, April 2007.
- [46] X. Rong Li and Vesselin P. Jilkov. Survey of maneuvering target tracking. part i: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, October 2003.
- [47] Robert A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-6(4):473–483, July 1970.
- [48] Robert A. Singer and Kenneth W. Behnke. Real-time tracking filter evaluation and selection for tactical applications. *IEEE Transactions on Aerospace and Electronic Systems*, AES-7(1):100–110, January 1971.
- [49] S. J. Asseo and R. J. Ardila. Sensor independent target state estimator design and evaluation. In *Proceedings of the National Aerospace and Electronics Conference (NAECON)*, pages 916–924, 1982.
- [50] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, April 1970.

- 
- [51] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- [52] Simon J. Julier, Jeffrey K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference, 1995.*, volume 3, pages 1628–1632 vol.3, 1995.
- [53] Simon J. Julier and Jeffrey K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical Report RRG, Department of Engineering Science, University of Oxford, 1996.
- [54] Simon J. Julier. The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference*, volume 6, pages 4555–4559, 2002.
- [55] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, volume Acquisition, Tracking and Pointing XI, pages 182–193, Orlando, Florida, USA, 1997.
- [56] Andreas Aristidou, Jonathan Cameron, and Joan Lasenby. Real-time estimation of missing markers in human motion capture. In *Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering, iCBBE'08*, Shanghai, China, May 2008.
- [57] Andreas Aristidou, Jonathan Cameron, and Joan Lasenby. Predicting missing markers to drive real-time centre of rotation estimation. In *Proceedings of the V Conference on Articulated Motion and Deformable Objects, AMDO'08*, volume 5098, pages 238–247, Mallorca, Spain, 2008. LNCS.