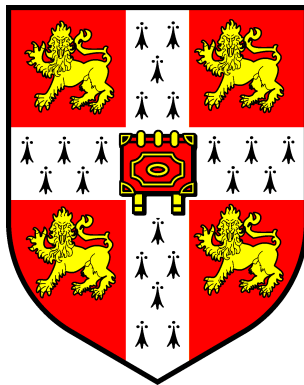


FIRST YEAR REPORT

REAL-TIME ESTIMATION OF MISSING DATA AND LOCALISATION OF THE CENTRE OF ROTATION IN OPTICAL MOTION CAPTURE DATA

Andreas Aristidou

aa462@cam.ac.uk



Cambridge University
Engineering Department
Signal Processing and Communication Group

in Division F, 2007/2008

Supervisor: Dr Joan Lasenby

April 7, 2008

© Andreas Aristidou

FIRST YEAR REPORT

REAL-TIME ESTIMATION OF
MISSING DATA AND LOCALISATION
OF THE CENTRE OF ROTATION IN
OPTICAL MOTION CAPTURE DATA

Andreas Aristidou

aa462@cam.ac.uk

Cambridge University
Engineering Department
Signal Processing and Communication Group

in Division F, 2007/2008

I, Andreas Aristidou hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signature:

Andreas Aristidou

Date: April 7, 2008

Abstract

This project considers the problem of taking 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 paper 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. Pre-defined models are not required and skeleton fitting to this complete data can then be updated in real-time.

Acknowledgements

This is a great opportunity to thank those who supported me through my first year experience in Cambridge. Firstly, I would like to thank Dr. Joan Lasenby, for making this study possible by providing me her regular supervision and support to conduct this project. I am also very grateful to my parents, who supported me during my postgraduate studies and for giving me the opportunity to follow my ambitions. Finally, I owe an enormous debt to Jonathan for his continued help and for introducing me to Geometric Algebra and quaternions and to Paris for his always useful comments and feedback.

Contents

Abstract	i
Acknowledgements	iii
Contents	v
List of Figures	ix
List of Tables	xi
1. Introduction	1
1.1. Optical Motion Capture	1
1.2. Literature Review and Motivation	1
1.3. Outline of the Report	2
2. Background	5
2.1. Geometric Algebra	5
2.1.1. The Products	5
2.1.2. Operators	9
2.1.3. The orthonormal basis for computation	10
2.1.4. Rotation using Rotors	11
2.1.5. Reflection	13
2.2. Conformal Geometric Algebra	14
2.2.1. Rotation	16
2.2.2. Translation	16
2.2.3. Dilation	17
2.2.4. Inversion	17
2.2.5. Blades in CGA	18
2.2.6. Intersections	25
2.3. Conclusions	29

3. Obtaining 3D marker positions	31
3.1. Apparatus	31
3.2. Marker-to-limb association	32
3.3. Rigid Body Dynamics	33
4. Real Time Joint Localisation	35
4.1. Introduction	35
4.2. Related Work	35
4.3. Finding the Rotor between 2 sets of vectors using Unit Quaternions	37
4.4. Finding the Centre of Rotation (CoR)	40
4.4.1. Cost function	40
4.4.2. Minimising the Cost function	41
5. Estimating the occluded markers	47
5.1. Related Work	48
5.2. Kalman Filter	49
5.3. Applying Constraints	51
5.3.1. All markers are visible on a given limb	51
5.3.2. One missing marker on a limb segment	52
5.3.3. Two missing markers on a limb segment	53
5.3.4. All markers on a limb segment are missing	53
5.3.5. Markers visible in only one camera	55
6. Results	57
6.1. The Experimental Environment	57
6.2. Results	57
6.2.1. One missing marker on a limb segment	58
6.2.2. Two missing markers on a limb segment	59
6.2.3. All markers on a limb segment are missing	59
6.2.4. Markers visible by one camera	60
6.3. Discussion	62
7. Conclusions and Future Work	69
A. Kalman Filter	73
A.1. An alternative approach	73
A.2. Kalman filter parameters	75
B. Finding the position of the occluded points using the rotors	77
C. Nearest point on circle from a point in space	79
C.1. The minimum distance from a circle and a point in space	79
C.2. The Solution	80

D. Nearest point on a line from a circle	83
E. Nearest point on a line from a sphere	87
List of Abbreviations	89
Bibliography	91

List of Figures

2.1.	The outer product	6
2.2.	The trivector	11
2.3.	The rotation effect of bivector B_3 over vectors e_1 and $e_1 + e_3$	11
2.4.	A reflection of the vector a in a plane perpendicular to m	13
2.5.	A line $L = P \wedge Q \wedge n$ passing through the null vectors P and Q and the ‘opposite’ line $L' = Q \wedge P \wedge n$	21
2.6.	A circle $L = P \wedge Q \wedge n$ passing through the null vectors P, Q and R	22
2.7.	The unit circle	22
2.8.	The 3-plane $\Phi = P \wedge Q \wedge R \wedge n$	24
2.9.	The solution for line to line intersection	26
3.1.	Marker-to-limb association example with three markers on each limb.	32
3.2.	Description of the rigid body.	33
4.1.	The motion of two limbs with a time-difference of m frames.	39
4.2.	A typical marker placement with relevant markers shown.	41
4.3.	The rotors during frames.	46
5.1.	The calculation of the observation vector in case where 2 markers are visible.	52
5.2.	The calculation of the observation vector in case of only one visible marker.	53
5.3.	The estimation procedure when all markers on a single limb segment are occluded.	54
5.4.	The observation vector in the case of 2 visible markers and one marker visible only by one camera.	55
6.1.	Examples of localisation of the CoR, where two non-visible markers exist (1 on each limb).	64
6.2.	The Kalman filter error.	65

6.3.	An example of the estimated and true position of markers and CoR location	66
6.4.	Examples of implementation on real data (Lower Body).	67
6.5.	Examples of implementation on real data (Arms).	68
B.1.	The calculation of the observation vector in the case where 2 markers are visible.	78
C.1.	Finding nearest point on circle to point in space using CGA/OCGA.	80
D.1.	Finding the nearest point on a line from a circle using CGA. Case where the distance from the centre of the circle C_1 to that point Y is smaller than the radius of the circle and point P is inside the circle area.	85
D.2.	Finding the nearest point on a line from a circle using CGA. Case where the distance from the centre of the circle C_1 to that point Y is smaller than the radius of the circle and point P is outside the circle area.	85
D.3.	Finding the nearest point on a line from a circle using CGA. Case where the distance from the centre of the circle C_1 to that point Y is greater than the radius of the circle and point P	86

List of Tables

2.1.	The orthonormal basis for computation	10
6.1.	Average results on real data with occlusions generated by deletions. Case of one missing marker on a limb segment.	58
6.2.	Average results on real data with occlusions generated by deletions. Case of two missing markers on a limb segment.	59
6.3.	Average results on real data with occlusions generated by deletions. Case where all markers on a limb segment are missing.	60
6.4.	Average results on real data with occlusions generated by deletions. Case of one missing marker on each limb segment.	61
6.5.	Average results on real data with occlusions generated by deletion. Case of one visible and two partially visible markers.	61
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.	62

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 make 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.

We propose a real-time approach for estimating the position of occluded markers using previous positions and information inferred from an approximate rigid body assumption. 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 markers¹ 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. It is assumed that the skeleton consists of rigid limbs connected with ball joints.

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 is a brief introduction to Geometric Algebra (GA) and to those aspects of GA that are used within this report. Also, Chap. 2 outlines an extension to Conformal Geometric Algebra (CGA) and examined how the Conformal Model can be represent geometric primitives such as points, lines, circles, planes and spheres.

Chapter 3 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.

¹Neighbouring markers are considered as markers belonging to the same limb segment.

The second section is concerned with the problem of fitting skeletal models to marker-based optical motion capture data. Chapter 4 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 occlusions by other limbs. Chapter 5 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 6 presents, compares and discusses the results of the methods proposed in Chapter 5.

Finally, Chapter 7 presents conclusions and suggestions for future work.

2

Background

This chapter presents a brief overview of Geometric Algebra (GA) in order to provide explanations of those elements used within this paper. More detailed treatment of geometric algebra can be found in [15, 16, 17].

2.1. Geometric Algebra

Classical vector algebra has a number of problems when we move from three dimensional space (Euclidean) to higher dimensional space. Hence, Hermann Grassmann (1809-77) and William Clifford (1845-79), attempted to create an ‘algebra of vectors’ in order to generalise conventional vector algebra to higher dimensions. The modern day extension of this work is now known as ‘Geometric Algebra’.

2.1.1. The Products

The geometric product is the most fundamental product of Geometric Algebra. However, it is important first to define the *inner* and *outer products* for vectors and then to introduce the geometric product of vectors.

The inner product

A vector space alone is insufficient for describing Euclidean geometry as it lacks the concepts of distance and angles. Distances and angles are important in order

to define entities like circles or perpendicular lines. Both can be defined through the introduction of a scalar product between vectors. This is known as the *inner product*, is written as $a \cdot b$ and returns a scalar. In Euclidean space the scalar product is always positive,

$$a^2 = a \cdot a > 0 \quad \forall a \neq 0 \quad (2.1)$$

Thus, the length of a vector $|a|$ can be defined as:

$$|a| = \sqrt{(a \cdot a)} \quad (2.2)$$

Hence, the inner product between a and b is defined via

$$a \cdot b = |a||b| \cos(\theta) \quad (2.3)$$

where θ is the angle between the vectors. The inner product can be also defined for higher dimensional generalisations of the vector (multivectors).

The outer product

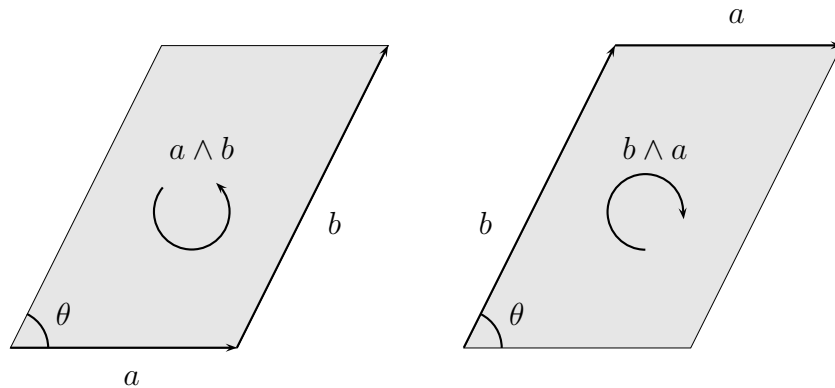


Figure 2.1.: The *outer product*. The outer or wedge product of a and b returns a directed area element of area $|a||b| \sin(\theta)$

The major failure of the cross product is that it exists only in 3 dimensions. In 2D there is nowhere else to go, whereas in more than 3 dimensions this direction is not uniquely defined. The solution of this problem was solved by Grassmann ([18]) encoding a plane geometrically, without relying on the notion of a vector perpendicular to it. Grassmann introduced a key feature of GA namely the *outer* or *exterior product*, which is written as $a \wedge b$. Unlike the cross product, which results in a perpendicular vector, the outer product of a pair of vectors operands results in a directed area, as shown in figure 2.1. This is the directed

area swept out by a and b and can be visualised as the parallelogram obtained by sweeping one vector along the other. Changing the order of the vectors reverses the orientation of the plane. The plane has area $|a||b|\sin(\theta)$, which is defined to be the magnitude of $a \wedge b$.

The outer product has the following key properties:

- The outer product of vectors is antisymmetric

$$a \wedge b = -b \wedge a \quad (2.4)$$

It follows that $a \wedge a = 0$.

- Also the outer product is distributive over addition

$$a \wedge (b + c) = a \wedge b + a \wedge c \quad (2.5)$$

- and associative

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c = a \wedge b \wedge c \quad (2.6)$$

- The outer product of a vector and a bivector defines a trivector that is an oriented volume.
- Although the formation of a bivector is often illustrated as the result of sweeping one vector along a second to form a parallelogram, the use of any particular shape is misleading as it is easy to show that the outer product of many different pairs of vectors will result in the same bivector.

The geometric product

William Clifford (1845-1879) made the next step and, investigating the work of Grassmann, he turned GA to a useful algebra. Clifford introduced the key feature of GA, the *geometric product*, in order to define a product that could identify the roles of the terms in a complex product. The geometric product can be expressed in terms of the inner and outer product, and it is defined as:

$$ab = a \cdot b + a \wedge b \quad (2.7)$$

The result of the geometric product seems strange, having two components lying in two different spaces, a scalar ($a \cdot b$) and a bivector ($a \wedge b$). This combination is referred to as a *multivector* and is analogous to complex numbers where we linearly real and imaginary numbers are combined to form the complex number.

Since,

$$ba = b \cdot a + b \wedge a = a \cdot b - a \wedge b \quad (2.8)$$

we can define the inner and outer product in terms of the symmetric and antisymmetric parts of the geometric product. Thus, for vectors a and b

$$\begin{aligned} a \cdot b &= \frac{1}{2}(ab + ba) \\ a \wedge b &= \frac{1}{2}(ab - ba) \end{aligned} \quad (2.9)$$

Then, the geometric product can be extended to an arbitrary number of vectors with the following properties:

- The geometric product is associative

$$(ab)c = a(bc) = abc \quad (2.10)$$

- The geometric product is distributive over addition

$$a(b + c) = ab + ac \quad (2.11)$$

- The symmetric part of the geometric product of two vectors is a scalar
- The geometric product has a unique inverse

$$ab \left(\frac{b}{b^2} \right) = a \quad (2.12)$$

hence, multiplying by $\frac{b}{b^2}$ is the inverse of multiplying by b .

The geometric product can also be used to define the inner and outer product of elements with single grades, A_r and B_s , where r and s denote the grade of the vector. Such elements are called blades¹.

$$A_r \cdot B_s = \langle A_r B_s \rangle_{|r-s|} \quad \text{if } r, s > 0 \quad (2.13)$$

$$A_r \cdot B_s = 0 \quad \text{if } r = 0 \text{ or } s = 0 \quad (2.14)$$

$$A_r \wedge B_s = \langle A_r B_s \rangle_{r+s} \quad (2.15)$$

where $\langle \rangle_t$ denotes the grade extraction operator, which sets all multivector components of grade other than t to zero. For the extraction of the scalar part of a multivector, the subscript 0 is usually dropped, and it is formulated as $\langle \rangle$ instead of $\langle \rangle_0$.

¹The term *blade* in GA is used to refer to quantities that can be written as the outer product of r -vectors, where r is termed the *grade*. For example, an r -grade can always be written as $A_1 \wedge A_2 \wedge \dots \wedge A_r$. More information about grades and the new orthonormal basis for computation can be found in section 2.1.3.

2.1.2. Operators

Geometric Algebra also has some other useful operators and elements. The most important of them are the *reverse* of a multivector, the *dual* and the *pseudoscalar*.

Reversion

Reversion substantially refers to the reversing of the order of any set of vectors that can be used to define a multivector, and it is symbolised as $\tilde{\cdot}$. A general example of reversion for the case of a $3d$ GA is given below:

$$\begin{aligned} M &= \alpha + a + B + \beta I \\ \tilde{M} &= \alpha + a - B - \beta I \end{aligned} \quad (2.16)$$

where M is multivector, α is a scalar, a is a vector, B is a bivector and βI is a trivector. Using the fact that $ab = -ba$, the reversion operator can be expressed by changing the sign of some particular grades of elements (bivectors and trivectors in $3d$) within the multivector.

The Pseudoscalar

The pseudoscalar refers to the highest grade basis element and it is symbolised as I_n , where n is the dimension of the space. For instance, in $3d$ GA the pseudoscalar can be written as:

$$I_3 = e_1 e_2 e_3 \quad (2.17)$$

where e_1, e_2, e_3 are the orthonormal basis vectors. The reversion of I_3 is equal to

$$\tilde{I}_3 = e_3 e_2 e_1 = -e_1 e_2 e_3 = -I_3 \quad (2.18)$$

The pseudoscalar is also denoted as a simple I , because the grade is always the highest and it is unnecessary to repeat it.

Since the pseudoscalar I is the unique right-handed unit trivector in the algebra, it gives us a number of new products. When we take the product of I with the vector e_i , we produce a bivector, eg.

$$I e_1 = e_1 e_2 e_3 e_1 = -e_1 e_2 e_1 e_3 = e_2 e_3 \quad (2.19)$$

$$e_1 I = e_1 e_1 e_2 e_3 = e_2 e_3 = I e_1 \quad (2.20)$$

Vectors therefore communicate with the $3d$ pseudoscalar,

$$I a = a I, \quad \text{for all } a \quad (2.21)$$

Hence, all the basis bivectors can be expressed as the product of the pseudoscalar and a *dual* vector.

$$\begin{aligned} Ie_1 &= e_1e_2e_3e_1 = e_2e_3 \\ Ie_2 &= e_1e_2e_3e_2 = -e_1e_3 = e_3e_1 \\ Ie_3 &= e_1e_2e_3e_3 = e_1e_2 \end{aligned}$$

The square of the pseudoscalar is equal to -1:

$$I^2 = e_1e_2e_3e_1e_2e_3 = e_1e_2e_3e_3e_1e_2 = e_1e_2e_1e_2 = -1 \quad (2.22)$$

Duality

Duality is another frequently used concept of Geometric Algebra. The dual transformation is denoted as $[\]^*$ and we map into the dual space by multiplying with the pseudoscalar as formulated in the equation 2.23.

$$[M]^* = MI \quad (2.23)$$

2.1.3. The orthonormal basis for computation

Although many results in GA can be reached without resorting to a basis, it is helpful to have a basis specifying the numerical multivectors for the 3d GA. Hence, the multivector basis can be defined via outer products of the three orthonormal basis vectors e_1 , e_2 and e_3 . The outer product is usually visualised geometrically as the movement of one vector along the other to form a ‘directed area’. This is a new object, neither a vector nor a scalar. It is termed a *bivector*. Similarly, the result of the outer product of this bivector and another vector is a *trivector*. Generally, a n -volume is termed an n -vector. Table 2.1 presents the multivector orthonormal basis.

Table 2.1.: The orthonormal basis for computation

Scalar:	a		
Vector:	e_1	e_2	e_3
Bivector:	$e_{12} = e_1 \wedge e_2$	$e_{13} = e_1 \wedge e_3$	$e_{23} = e_2 \wedge e_3$
Trivector:	$e_{123} = e_1 \wedge e_2 \wedge e_3$		

We can say that a scalar is grade 0, a vector is grade 1, a bivector is grade 2, as it is formed from 2 vectors, etc. Generally, a n -vector has grade n . Figure 2.2 illustrates an example of a trivector (sweeping $a \wedge b$ along c).

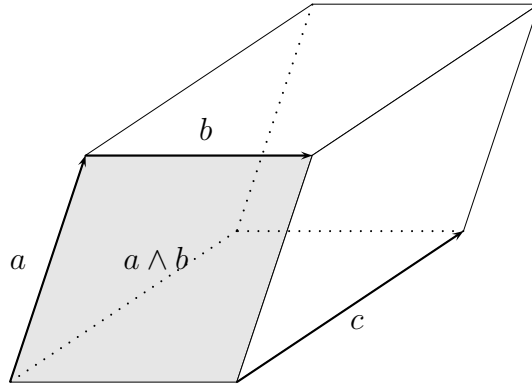


Figure 2.2.: The trivector or directed volume is the result of sweeping $a \wedge b$ along c .

2.1.4. Rotation using Rotors

Consider any three orthonormal basis vectors of \mathbb{R}^3 , $\{e_1, e_2, e_3\}$, then the new three basis bivectors generated are the $B_1 = e_2e_3$, $B_2 = e_3e_1$ and $B_3 = e_1e_2$. The basis bivectors all square to -1 , and all anticommute as given below:

$$B_i^2 = e_i e_j e_i e_j = -e_i e_j e_j e_i = -1 \quad \text{for } i \neq j$$

An important feature of the bivectors is their effect on vectors. For example, the effect of the bivector B_3 on the vectors e_1 and $e_1 + e_2$ is to rotate them counter-clockwise by 90 degrees. This example is illustrated in figure 2.3.

$$\begin{aligned} e_1 B_3 &= e_1 e_1 e_2 = e_2 \\ e_2 B_3 &= e_2 e_1 e_2 = -e_2 e_2 e_1 = -e_1 \\ (e_1 + e_2) B_3 &= e_1 B_3 + e_2 B_3 = e_2 - e_1 \end{aligned}$$

This is applicable for every bivector $e_i e_j$ over the plane defined by e_i and e_j . It is important here to mention that this method also works in higher-dimension spaces.

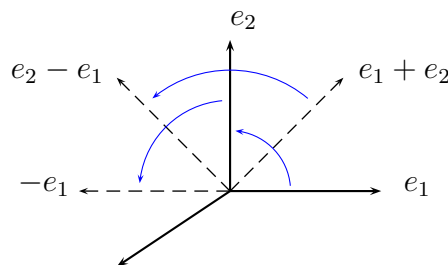


Figure 2.3.: The rotation effect of bivector B_3 over vectors e_1 and $e_1 + e_3$.

Thus, any vector a in the plane defined by e_1 and e_2 can be represented using the formula:

$$\begin{aligned} a &= r (e_1 \cos \theta + e_2 \sin \theta) \\ &= e_1 r (\cos \theta + B_3 \sin \theta) \end{aligned} \quad (2.24)$$

where r is the distance of the point a from the origin and θ is the angle between a and e_1 . This can be generalised if, instead of B_3 , the unit bivector $\hat{B} = \frac{a \wedge b}{|a \wedge b|}$ is used. Hence, the rotation in the plane described by \hat{B} by an angle 2θ , can be expressed as:

$$\begin{aligned} R &= \exp(\theta \hat{B}) \\ &= 1 + \frac{\theta \hat{B}}{1!} + \frac{\theta^2 \hat{B}^2}{2!} + \frac{\theta^3 \hat{B}^3}{3!} + \dots \\ &= \cos \theta + \hat{B} \sin \theta \end{aligned} \quad (2.25)$$

Then, any vector a which lies in the plane of the bivector \hat{B} can be represented by:

$$a = e_1 r \exp(\theta \hat{B}) \quad (2.26)$$

Using this, it can be showed that the rotation by ϕ radians in the plane of \hat{B} is accomplished by:

$$a \mapsto a' = a \exp(\phi \hat{B}) = a (\cos(\phi) + \hat{B} \sin(\phi)) \quad (2.27)$$

It is important to mention that here the vector to be rotated a lies in the plane of rotation. In a different case, the vector should be decomposed into a component that lies in the plane, a_{\parallel} , and one normal to the plane, a_{\perp} .

$$a = a_{\parallel} + a_{\perp} \quad (2.28)$$

Now, assuming that $R = \exp\left(-\frac{\phi}{2} \hat{B}\right)$ and $\tilde{R} = \exp\left(\frac{\phi}{2} \hat{B}\right)$, consider the following operation:

$$\begin{aligned} a' &= R (a_{\parallel} + a_{\perp}) \tilde{R} \\ &= \left(\cos\left(\frac{\phi}{2}\right) - \hat{B} \sin\left(\frac{\phi}{2}\right) \right) (a_{\parallel} + a_{\perp}) \left(\cos\left(\frac{\phi}{2}\right) + \hat{B} \sin\left(\frac{\phi}{2}\right) \right) \\ &= a_{\perp} + \left(\cos^2\left(\frac{\phi}{2}\right) - \sin^2\left(\frac{\phi}{2}\right) \right) a_{\parallel} + 2 \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\phi}{2}\right) \hat{B} a_{\parallel} \\ &= a_{\perp} + a_{\parallel} (\cos(\phi) + \hat{B} \sin(\phi)) \end{aligned} \quad (2.29)$$

Equation 2.29 relies on $\hat{B}a_{\perp} = a_{\perp}\hat{B}$ and $\hat{B}a_{\parallel} = -a_{\parallel}\hat{B}$, which states that a bivector commutes with a perpendicular vector and anticommutes with a parallel vector.

Thus, the component of the vector that lies in the plane is rotated around an axis normal to the plane without affecting the component that is normal to the plane. This leads to a general method of rotation in any plane. Therefore, a given rotation ϕ in a plane specified by \hat{B} can be performed as:

$$a \mapsto Ra\tilde{R} \quad (2.30)$$

using the element $R = \exp\left(-\hat{B}\frac{\phi}{2}\right)$. R is referred to as a *rotor*.

The most important properties of the rotors are that $R\tilde{R} = 1$ and thus $\tilde{R} = R^{-1}$, and $Ra\tilde{R} = (-R)a(-\tilde{R})$, which means that there is a double covering of the space of rotations.

2.1.5. Reflection

Consider a reflection of the multivector a in the plane orthogonal to a unit r -vector m , where $m^2 = 1$, and r is the grade of the blade.

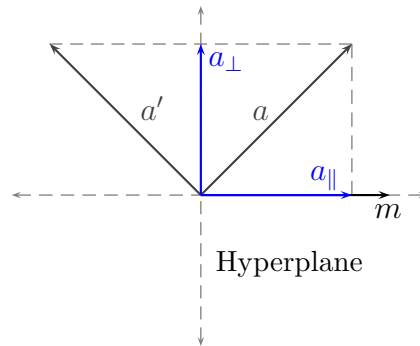


Figure 2.4.: A reflection of the vector a in a plane perpendicular to m . It is obvious that $a = a_{\perp} + a_{\parallel}$ and $a' = a_{\perp} - a_{\parallel}$

The component of a that is parallel to the plane m_r changes sign, whereas the perpendicular components remains unaffected, as showed in figure 2.4. Hence, the parallel component is the projection to m and the perpendicular component is the remainder.

$$a_{\parallel} = a \cdot mm \quad (2.31)$$

$$a_{\perp} = a - a \cdot mm = (am - a \cdot m)m = a \wedge mm \quad (2.32)$$

The result of the reflection is therefore:

$$\begin{aligned}
a' &= a_{\perp} - a_{\parallel} \\
&= -a \cdot mm + a \wedge mm \\
&= -(m \cdot a + m \wedge a)m \\
&= -mam
\end{aligned} \tag{2.33}$$

2.2. Conformal Geometric Algebra

This section is an introduction to the Conformal Model of Geometric Algebra (CGA) [19] as first introduced by Hestenes and Sobczyk in 1984. CGA extend the space from 3 to 5 dimensions by adding two additional basis vectors providing additional tools with which to represent and manipulate geometry. Hence, the additional basis vectors e and \bar{e} are added to the 3 existing basis vectors e_1 , e_2 and e_3 of the 3d GA. e and \bar{e} are defined as:

$$e^2 = +1 \qquad \bar{e}^2 = -1 \tag{2.34}$$

The new basis vectors provides a space that allows null vectors, n and \bar{n} , which are defined as:

$$n = e + \bar{e} \qquad \bar{n} = e - \bar{e} \tag{2.35}$$

where n is associated with the point at infinity and \bar{n} with the origin. It is very easy to prove that these vectors are null since:

$$\begin{aligned}
n^2 &= (e + \bar{e}) \cdot (e + \bar{e}) \\
&= e^2 + 2(e \cdot \bar{e}) + \bar{e}^2 \\
&= 1 + 0 - 1 = 0
\end{aligned} \tag{2.36}$$

and

$$\begin{aligned}
\bar{n}^2 &= (e - \bar{e}) \cdot (e - \bar{e}) \\
&= e^2 - 2(e \cdot \bar{e}) + \bar{e}^2 \\
&= 1 - 0 - 1 = 0
\end{aligned} \tag{2.37}$$

Two basic identities for n and \bar{n} are:

$$n \cdot \bar{n} = (e + \bar{e}) \cdot (e - \bar{e}) = e^2 - \bar{e}^2 = 2 \tag{2.38}$$

$$x \cdot n = x \cdot \bar{n} = 0 \tag{2.39}$$

where $x \in \mathbb{R}^5$. Another observation is that the squared bivector E^2 , where $E = n \wedge \bar{n}$ is equal to 4. This proved below:

$$\begin{aligned} E^2 &= (n \wedge \bar{n}) \cdot (n \wedge \bar{n}) \\ &= (n \cdot \bar{n})(n \cdot \bar{n}) - n^2 \bar{n}^2 \\ &= 4 \end{aligned} \tag{2.40}$$

since $n \cdot \bar{n} = 2$ and $n^2 = \bar{n}^2 = 0$.

The Hestenes' mapping

The mapping used in Hestenes' illustration ([19], page 302) is used to take a spatial (conventional 3d) vector to the equivalent CGA representation. Thus,

$$H(x) = \frac{1}{2}(x - e)n(x - e) \tag{2.41}$$

Substituting for $n = e + \bar{e}$ and using the fact that $\bar{e} \cdot x = \bar{e} \cdot e = n \cdot x = 0$, it is possible to rewrite the equation in terms of null vectors as:

$$H(x) = \frac{1}{2}(x^2 n + 2x - \bar{n}) \tag{2.42}$$

We can show that $H(x)$ is always a null vector by evaluating:

$$\begin{aligned} [H(x)]^2 &= \frac{1}{4}(x^2 n + 2x - \bar{n}) \cdot (x^2 n + 2x - \bar{n}) \\ &= -\frac{1}{2}x^2 n \cdot \bar{n} + x^2 \\ &= -x^2 + x^2 = 0 \end{aligned} \tag{2.43}$$

as $\bar{e} \cdot x = \bar{e} \cdot e = n \cdot x = 0$. Another interesting observation is that, for any vector A and $B \in \mathbb{R}^5$ which represent the points a and b in \mathbb{R}^4 , $A \cdot B$ is related to the Euclidean distance between the points a and b , as proved in equation 2.44.

$$\begin{aligned} A \cdot B &= H(a) \cdot H(b) \\ &= \frac{1}{4}(a^2 n + 2a - \bar{n}) \cdot (b^2 n + 2b - \bar{n}) \\ &= -\frac{1}{2}a^2 + a \cdot b - \frac{1}{2}b^2 \\ &= -\frac{1}{2}(a - b)^2 \end{aligned} \tag{2.44}$$

Hence, the Euclidean distance between the two points a and b can be defined as:

$$d(A, B) = \sqrt{-2(A \cdot B)} \tag{2.45}$$

More information about *distance geometry* can be found in [20], [19] and [21].

2.2.1. Rotation

In Conformal Geometric Algebra rotations are also performed with the rotor elements as in usual Geometric Algebra (see equation 2.29). Thus, having in mind that $R\bar{n}\tilde{R} = \bar{n}$ and $Rn\tilde{R} = n$, $x \mapsto Rx\tilde{R}$ can be replaced by $H(x) \mapsto H(Rx\tilde{R})$, as is shown below.

$$\begin{aligned}
 RH(x)\tilde{R} &= R\frac{1}{2}(x^2n + 2x - \bar{n})\tilde{R} \\
 &= \frac{1}{2}(x^2Rn\tilde{R} + 2Rx\tilde{R} - R\bar{n}\tilde{R}) \\
 &= \frac{1}{2}(x^2n + 2Rx\tilde{R} - \bar{n}) \\
 &= H(Rx\tilde{R})
 \end{aligned} \tag{2.46}$$

2.2.2. Translation

The translation along a vector a is defined as the mapping $x \mapsto x + a$ for x . In this section it will be proved that this can be performed by applying a rotor $R = T_a = \exp(\frac{na}{2})$. Using the power series expansion of the exponential, the rotor can be simplified to:

$$T = \exp\left(\frac{na}{2}\right) = 1 + \frac{na}{2} + \frac{1}{2}\left(\frac{na}{2}\right)^2 + \dots = 1 + \frac{na}{2} \tag{2.47}$$

since n is null, $an = -na$ and therefore the higher order terms are all zero.

The rotor T acts over the vectors n , \bar{n} and x as below:

$$\begin{aligned}
 Tn\tilde{T} &= \left(1 + \frac{na}{2}\right)n\left(1 + \frac{an}{2}\right) \\
 &= n + nan + \frac{nanan}{4} \\
 &= n
 \end{aligned} \tag{2.48}$$

Similarly,

$$\begin{aligned}
 T\bar{n}\tilde{T} &= \left(1 + \frac{na}{2}\right)\bar{n}\left(1 + \frac{an}{2}\right) \\
 &= \bar{n} + \frac{na}{2}\bar{n} + \bar{n}\frac{an}{2} + \frac{na}{2}\bar{n}\frac{an}{2} \\
 &= \bar{n} - 2a - a^2n
 \end{aligned} \tag{2.49}$$

and

$$\begin{aligned}
 Tx\tilde{T} &= \left(1 + \frac{na}{2}\right)x\left(1 + \frac{an}{2}\right) \\
 &= x + n(a \cdot x)
 \end{aligned} \tag{2.50}$$

Therefore, the translation over the null vector $H(x)$ will be:

$$\begin{aligned}
TH(x)\tilde{T} &= \left(1 + \frac{na}{2}\right) \frac{1}{2} (x^2n + 2x - \bar{n}) \left(1 + \frac{an}{2}\right) \\
&= \frac{1}{2} (x^2n + 2(x + n(a \cdot x)) - (\bar{n} - 2a - a^2n)) \\
&= \frac{1}{2} ((x + a)^2n + 2(x + a) - \bar{n}) \\
&= \frac{1}{2} (x^2n + 2x - \bar{n}) = H(x + a)
\end{aligned} \tag{2.51}$$

To summarise, the $x \mapsto x + a$ can be replaced by $H(x) \mapsto TH(x)\tilde{T} = H(x + a)$.

2.2.3. Dilation

A dilation by a factor of α can be represented by mapping $x \mapsto \alpha x$. This can be achieved by considering the rotor $R = D_a = \exp\left(\frac{\alpha}{2}e\bar{e}\right)$. Since $-e\bar{e}n = ne\bar{e} = n$ and $-\bar{n}e\bar{e} = e\bar{e}\bar{n} = \bar{n}$, the dilation of the vector $H(x)$ by a factor of $\exp(-\alpha)$ about the origin returns:

$$\begin{aligned}
DH(x)\tilde{D} &= \exp\left(\frac{\alpha}{2}e\bar{e}\right) \frac{1}{2} (x^2n + 2x - \bar{n}) \exp\left(-\frac{\alpha}{2}e\bar{e}\right) \\
&= \frac{1}{2} \left(x^2 \exp\left(\frac{\alpha}{2}e\bar{e}\right) n + 2x - \exp\left(\frac{\alpha}{2}e\bar{e}\right) \bar{n}\right) \\
&= \frac{1}{2} (x^2 \exp(-\alpha) n + 2x - \exp(\alpha) \bar{n}) \\
&= \exp(\alpha) \frac{1}{2} (\exp(-2\alpha)x^2n + 2\exp(-\alpha)x - \bar{n}) \\
&= \exp(\alpha) \frac{1}{2} (x^2n + 2x - \bar{n}) = H(\exp(-\alpha)x)
\end{aligned} \tag{2.52}$$

Hence, the dilation $x \mapsto \exp(-\alpha)x$ can be represented by $H(x) \mapsto DH(x)\tilde{D} = \exp(\alpha)H(\exp(-\alpha)x)$.

2.2.4. Inversion

Inversion in the origin corresponds to the mapping $x \mapsto \frac{x}{x^2}$ or for non-singular vectors $x \mapsto x^{-1}$. This can be performed by reflecting in e . The reflections in e of the vectors n , \bar{n} and x are given by:

$$\begin{aligned}
-ene &= -e\bar{e}\bar{n} = -\bar{n} \\
-e\bar{n}e &= -e\bar{e}n = -n \\
-exe &= x
\end{aligned}$$

The inversion of $H(x)$ under the reflection in e then gives

$$\begin{aligned}
-eH(x)e &= -e\frac{1}{2}(x^2n + 2x - \bar{n})e \\
&= \frac{1}{2}(-x^2\bar{n} + 2x + n) \\
&= x^2\frac{1}{2}\left(\frac{1}{x^2}n + 2\frac{x}{x^2} - \bar{n}\right) \\
&= x^2H\left(\frac{x}{x^2}\right)
\end{aligned} \tag{2.53}$$

Thus, the inversion $x \mapsto \frac{x}{x^2}$ is replaced by $H(x) \mapsto -\frac{eH(x)e}{x^2} = H\left(\frac{x}{x^2}\right)$.

2.2.5. Blades in CGA

The term *blade* in GA is used to refer to quantities that can be written as the wedge product of vectors. For example, an r -blade can always be written as $A_1 \wedge A_2 \wedge \dots \wedge A_r$, which differs from an r -vector that may be any linear combination of r -blades. In this report, it is assumed that all null-vectors X are defined such that $X \cdot n = -1$, unless otherwise stated.

Vectors and 2-blades

A 2-blade is formed from the outer product $A \wedge B$ of two null vectors A and B . Considering the differentiation between signs of null vectors in CGA, many different separations are possible, such:

$$A \wedge B = -(B \wedge A) = -B \wedge A = B \wedge -A \tag{2.54}$$

The outer product $A \wedge B$ can be separated into a pair of individual null-vectors that are unique up to a scale, as shown below.

Extracting vector A and vector B from $A \wedge B$

We often need to extract the two vectors A and B from the bivector $A \wedge B$. This paragraph presents a solution of this problem using *projectors*. Assume that A and B are normalised so that $A \cdot n = -1$ and $B \cdot n = -1$. Let the 2-blade $T = A \wedge B$ and form

$$F = \frac{1}{\beta}A \wedge B \tag{2.55}$$

where $\beta > 0$ and $\beta^2 = T^2$, so that $F^2 = 1$ if $\beta^2 \neq 0$. Thus, the two projector operators can be defined as:

$$\begin{aligned}
P &= \frac{1}{2}(1 + F) \\
\tilde{P} &= \frac{1}{2}(1 - F)
\end{aligned} \tag{2.56}$$

where \tilde{P} denotes the normal reversion operation applied to P . Note that $PP = P$, which can be verified

$$\begin{aligned} PP &= \frac{1}{4} (1 + F) (1 + F) \\ &= \frac{1}{4} (1 + 2F + 1) \\ &= \frac{1}{2} (1 + F) \end{aligned} \quad (2.57)$$

Similarly, it can be shown that $\tilde{P}\tilde{P} = \tilde{P}$. An equally important property is that $\tilde{P}P = P\tilde{P} = 0$ which, again, is easy to show

$$P\tilde{P} = \frac{1}{4} (1 + F) (1 - F) = \frac{1}{4} (1 - F^2) = \frac{1}{4} (1 - 1) = 0 \quad (2.58)$$

In the same way, it can be shown that $\tilde{P}P = 0$.

It is important here to show what effect these projectors have on A and B . Hence, P and \tilde{P} acting on A and B will return the following results:

$$\begin{aligned} PA &= 0 \\ PB &= B \\ \tilde{P}A &= A \\ \tilde{P}B &= 0 \end{aligned}$$

This can be easily verified as below:

$$\begin{aligned} PA &= \frac{1}{2} \left[1 + \frac{1}{\beta} A \wedge B \right] A \\ &= \frac{1}{2} \left[A + \frac{1}{\beta} (A \wedge B) A \right] \\ &= \frac{1}{2} \left[A + \frac{1}{\beta} (A \cdot B) A \right] \\ &= \frac{1}{2} (A - A) = 0 \end{aligned} \quad (2.59)$$

since

$$\begin{aligned} (A \wedge B) A &= (A \wedge B) \cdot A \\ &= -A^2 B + (A \cdot B) A \\ &= (A \cdot B) A, \text{ because } A^2 = 0 \end{aligned}$$

and

$$A \cdot B = -\beta \quad (2.60)$$

This follows from equation 2.45, where $A \cdot B$ must be negative, the facts that $A^2 = B^2 = 0$ and also from:

$$\begin{aligned}\beta^2 &= (A \wedge B) \cdot (A \wedge B) \\ &= -A^2 B^2 (A \cdot B)^2\end{aligned}\quad (2.61)$$

The next step is to consider the vector obtained by dotting $A \wedge B$ with n .

$$\begin{aligned}(A \wedge B) \cdot n &= -n \cdot (A \wedge B) \\ &= -(n \cdot A) B + (n \cdot B) A \\ &= (B - A)\end{aligned}\quad (2.62)$$

using the fact that A and B are normalised points such that $A \cdot n = B \cdot n = -1$. Thus, it follows that:

$$P[(A \wedge B) \cdot n] = P(B - A) = B \quad (2.63)$$

$$-\tilde{P}[(A \wedge B) \cdot n] = -\tilde{P}(B - A) = A \quad (2.64)$$

It is also noted that since $AP = \tilde{P}A = A$ it follows that $\tilde{P}AP = \tilde{P}\tilde{P}A = \tilde{P}A$. In that way we have:

$$\begin{aligned}\tilde{P}AP &= \tilde{P}A \\ PA\tilde{P} &= 0 \\ PB\tilde{P} &= PB \\ \tilde{P}BP &= 0\end{aligned}$$

which mean that the projectors can be written as two-sided operations. Hence, the two points A and B can be extracted from a 2-blade $A \wedge B$ as:

$$A = -\tilde{P}[(A \wedge B) \cdot n] \equiv -\tilde{P}[(A \wedge B) \cdot n] P = -\tilde{P}[\langle (A \wedge B) n \rangle_1] \quad (2.65)$$

$$B = P[(A \wedge B) \cdot n] \equiv P[(A \wedge B) \cdot n] \tilde{P} = P[\langle (A \wedge B) n \rangle_1] \quad (2.66)$$

The use of projectors ensures that one projector, P , will always return the first point, A , and the second projector, \tilde{P} , will return the second point, B . As a result, an orientation may be inferred for this point pair, hence one is considered to occur before the other. However, it not possible to extract the null vector from a bivector having one of its points as n (e.g. $B = A \wedge n$) using the 2-blade projector-based separation method. One approach to get round this problem was proposed in [22] and in [23] with minor corrections. A bivector B may be in the form $A \wedge n$ or $n \wedge A$, where A is the null vector representing the spatial intersection point a . Hence,

$$a = \frac{1}{4} (B \wedge \bar{n}) \cdot N \quad \Leftrightarrow n \text{ is the first component} \quad (2.67)$$

$$a = \frac{1}{4} (\bar{n} \wedge B) \cdot N \quad \Leftrightarrow n \text{ is the second component} \quad (2.68)$$

where $N = n \wedge \bar{n}$.

The usefulness of these results will become palpable later when intersections are considered (section 2.2.6).

3-Vectors

In this section we study trivectors. Consider the null vectors P , Q and R in the $5d$ space representing the points p , q and r respectively in $3d$ space .

Lines as trivectors In order to define a line with direction p to q , the trivector L needs to be introduced. The trivector L is formed as:

$$\begin{aligned} L &= P \wedge Q \wedge n & (2.69) \\ &= Q \wedge n \wedge P \\ &= n \wedge P \wedge Q \end{aligned}$$

and corresponds to a line that is passing through the points represented by the null vectors P and Q and the point at infinity, n . It can be shown algebraically that these 3 lines returns the same blade. However, the line $L' = Q \wedge P \wedge n = -L$ will have the opposite direction, differing only in sign, due to the anti-commuting nature of the outer product of the vectors. An example of the lines L and L' is given in figure 2.5.

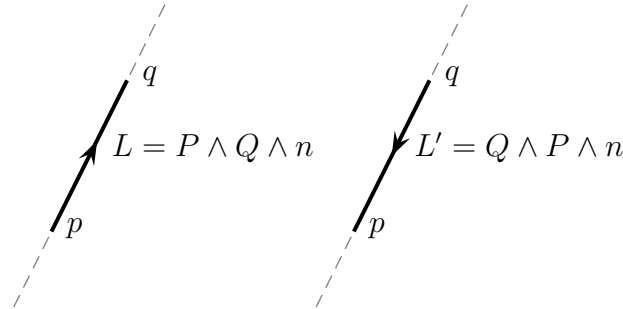


Figure 2.5.: A line $L = P \wedge Q \wedge n$ passing through the null vectors P and Q and the ‘opposite’ line, $L' = Q \wedge P \wedge n$.

Circles as trivectors A circle can be defined as a trivector C , where the point at infinity, n , is replaced by a third point on the circle, R . Hence, the equation of a circle in CGA is formed as:

$$C = P \wedge Q \wedge R \quad (2.70)$$

where C corresponds to a circle that is passing through the points represented by the null vectors P , Q and R , as shown in figure 2.6.

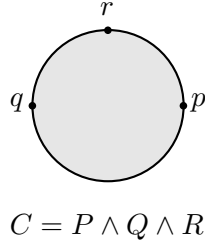


Figure 2.6.: A circle $L = P \wedge Q \wedge n$ passing through the null vectors P , Q and R .

The unit circle (figure 2.7) in the plane is formed as the circle passing through the three key points, e_1 , $-e_1$ and e_2 . Hence, for any unit length vector x , we know that $H(x) = \frac{1}{2}(n + 2x - \bar{n}) = (x + \bar{e})$. Thus, we have:

$$H(e_1) \wedge H(e_2) \wedge H(-e_1) = 2e_1e_1\bar{e}$$

and hence the trivector $C = 2e_1e_1\bar{e}$ represents the unit circle. The dual of the unit circle C^* is equal to

$$C^* = CI_4 = 2e = (n + \bar{n}) \quad (2.71)$$

where I_4 is the pseudoscalar given by $I_4 = e_1e_2e\bar{e}$.

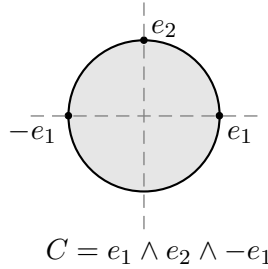


Figure 2.7.: The unit circle.

If X lies on the circle, we know that $X \wedge C = 0$. Since C^* is the dual of a trivector in a $4D$ space, it is a vector. Hence, an alternative but useful representation of the circle is given by

$$X \cdot C^* = 0$$

In the equation 2.44 we proved that for any two normalised point representations A and B , $A \cdot B = \frac{1}{2}(a - b)^2$. Thus, consider a point on circle, X , and the circle centre, B , then the radius ρ of the circle can be calculated as:

$$X \cdot B = \frac{1}{2}(x - b)^2 \equiv -\frac{1}{2}\rho^2 \quad (2.72)$$

For a normalised point representation X this implies that

$$X \cdot \left(B - \frac{1}{2}\rho^2 n \right) = 0 \quad (2.73)$$

since $X \cdot n = -1$. Comparing this with $X \cdot C^*$ we see that our normalised C^* is

$$C^* = B - \frac{1}{2}\rho^2 n \quad (2.74)$$

Therefore, the vector C^* encodes in a neat fashion the centre and the radius of the circle in the plane. We also note that the radius of the circle can be easily calculated by squaring C^*

$$\begin{aligned} (C^*)^2 &= \left(B - \frac{1}{2}\rho^2 n \right)^2 \\ &= -\rho^2 B \cdot n = \rho^2 \end{aligned} \quad (2.75)$$

since $B^2 = 0$, $n^2 = 0$ and $B \cdot n = -1$. Using equation 2.75 it is easy to show that:

$$B = C^* + \frac{1}{2}(C^*)^2 n \quad (2.76)$$

Note that the above relations assume that C^* is normalised such that $C^* \cdot n = -1$ since $C^* \cdot n = B \cdot n = -1$, as it is assumed that B is a normalised null vector.

However, there is a more elegant way of calculating the centre of a circle in $3d$, as proved in [22]. The centre of a circle, C , is also given by reflecting the point at infinity, n , in the circle as:

$$B = CnC \quad (2.77)$$

4-Vectors

This section studies 4-vectors. 4-vectors in $5d$ space represent 3-planes, Φ , and spheres, Σ .

Planes as 4-vectors A 3-plane, Φ , passing through the three points defined by null vectors P , Q and R is given by:

$$\Phi = P \wedge Q \wedge R \wedge n \quad (2.78)$$

Figure 2.8 illustrates an example of a 3-plane passing through the points p , q and r .

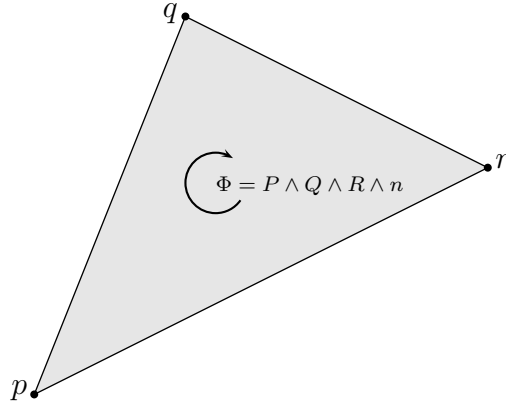


Figure 2.8.: The 3-plane $\Phi = P \wedge Q \wedge R \wedge n$.

Spheres as 4-vectors Spheres can be defined as a 4-vector where the point at the infinity, n , is replaced by a point on the sphere. Hence, given any 4 points p , q , r and s whose $5D$ representations are the null vectors P , Q , R and S respectively, the sphere passing through those points is defined by the 4-vector Σ

$$\Sigma = P \wedge Q \wedge R \wedge S \quad (2.79)$$

We know that $X \wedge \Sigma = 0$ for any X lying on the sphere. This can be rewritten as

$$X \cdot \Sigma^* = 0 \quad (2.80)$$

where Σ^* is the dual to Σ , hence is a vector. As for the circle solution, we can show that the dual representation of the sphere naturally encodes the centre and the radius of the sphere. Thus

$$X \cdot C = -\frac{1}{2}(x - c)^2 \equiv -\frac{1}{2}\rho^2 \quad (2.81)$$

where X is a point on a sphere, C is the centre of the sphere and ρ is the radius of the sphere. For a normalised point X , where $X \cdot n = -1$ this means that

$$X \cdot \left(C - \frac{1}{2}\rho^2 n \right) = 0 \quad (2.82)$$

Comparing equation 2.82 with equation 2.80, we can find that

$$\Sigma^* = C - \frac{1}{2}\rho^2 n \quad (2.83)$$

Thus, the dual vector Σ^* encodes the centre and radius of the sphere. The radius and the centre can immediately be calculated by squaring Σ^*

$$(\Sigma^*)^2 = \left(C - \frac{1}{2}\rho^2 n \right)^2 = \rho^2$$

since $C^2 = 0$, $n^2 = 0$ and $C \cdot n = -1$. Using this equation, we can easily show that:

$$C = \Sigma^* + \frac{1}{2}(\Sigma^*)^2 n \quad (2.84)$$

As for the circle case, the centre, C , of a sphere, Σ , is also given by reflecting the point at infinity, n , in the sphere as:

$$C = \Sigma n \Sigma \quad (2.85)$$

5-Vectors

5-vectors, such as the pseudoscalar I_5 , have two orientations and these correspond to Stolfi's concept [24] of an oriented universe. The pseudoscalar is defined as $I_5 = e_1 \wedge e_2 \wedge e_3 \wedge e \wedge \bar{e} = e_{12345}$, and it satisfies $I_5^2 = -1$.

2.2.6. Intersections

This section outlines the various ways of intersecting objects within the conformal geometric algebra model. In this paper, an operator termed *the meet* [19] is used, represented by the symbol \vee , which given two objects A and B returns their intersection.

Intersecting lines with lines

This paragraph considers the intersection of two lines. Let the lines L_1 and L_2 . The meet of these two lines is given by

$$X = L_1 \vee L_2 = [\langle L_1 L_2 \rangle_{2n-r-s}]^* \quad (2.86)$$

where n denotes the dimension in space, r is the grade of the first element and s the grade of the second. Thus, $2n - r - s = 10 - 3 - 3 = 4$, so the dual object has grade 1. However, if the lines intersect at a point, the meet, X , will not return this intersection point. Instead of that, if the lines intersect, then $L_1 \vee L_1 = 0$ and if the lines do not intersect, then $L_1 \vee L_1 \propto n$. In order to find the intersection point, we use the following procedure. Assume that the lines L_1 and L_2 intersect at point P . Reflect the line L_1 in line L_2 and get $L'_1 = L_2 L_1 L_2$. Then, find the line which is perpendicular to L_2 , passing through the point P , which is equal to $L_2^P = L_1 - L_2 L_1 L_2$. Take any arbitrary point representation Y and reflect in L_2^P via $Y' = L_2^P Y L_2^P$. Then, take the midpoint of Y and Y' , $Y'' = \frac{1}{2}(Y + Y')$, which must lie on the line L_2^P . Thereafter, reflect the point Y'' in line L_2 to give $Y''' = L_2 Y'' L_2$ and again take the midpoint $P' = \frac{1}{2}(Y'' + Y''')$. The representation of the intersection point can then be extracted via

$$P = \frac{-(P' n P')}{2(P' \cdot n)^2} \quad (2.87)$$

Figure 2.9 illustrates the above.

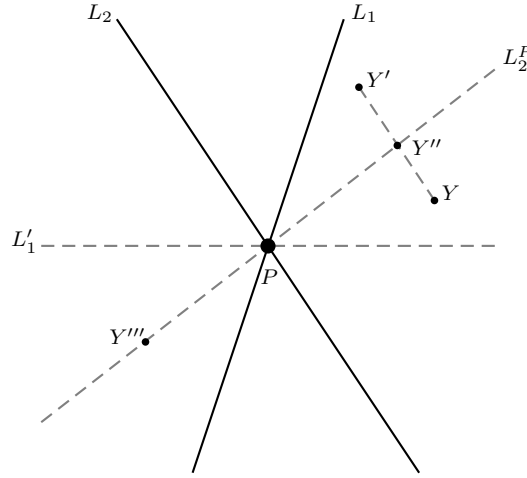


Figure 2.9.: The solution for line to line intersection.

Intersecting circles with circles and lines

The intersection of two circles or a circle and a line is the subject of this paragraph. Firstly, we will focus on the meet of two circles, C_1 and C_2 .

$$X = C_1 \vee C_2 = [\langle C_1 C_2 \rangle_{2n-r-s}]^* \quad (2.88)$$

where $2n - r - s = 10 - 3 - 3 = 4$, so that the dual object has grade 1. However, the intersection of two circles mostly returns two intersection points (when they lie in the same plane), and the 1-grade object X can not give us the two points. In that case, the meet does not return the intersection points but just an object that helps us to conclude whether the circles intersect at two points, a point or if they do not intersect at all. Hence,

$$\begin{aligned} \text{if circles have two intersections,} & \quad C_1 \vee C_2 = 0, \\ \text{if circles have one intersection,} & \quad C_1 \vee C_2 = X, \text{ where } X^2 = 0 \\ \text{if circles have no intersection,} & \quad C_1 \vee C_2 = X, \text{ where } X^2 \neq 0 \end{aligned}$$

If $C_1 \vee C_2 = X$ and $X^2 = 0$, then it is obvious that the intersecting point is represented by X . If the meet gives zero, there are two intersections, and these can easily be found by intersecting the plane of one of the circles with the other circle as:

$$B = C_1 \vee (C_2 \wedge n) = [\langle C_1 (C_2 \wedge n) \rangle_{2n-r-s}]^* \quad (2.89)$$

where $2n - r - s = 10 - 3 - 4 = 3$, so that the dual object has grade 2. Thus, the two points can be extracted from the bivector B using the projectors given in equation 2.65.

Instead of having intersection of two circles, consider the intersection of a circle C_1 and a line L_1 . The meet is again a grade 1 object, and it similarly holds that:

- if circle and line have two intersections, $C_1 \vee L_1 = 0$,
- if circle and line have one intersection, $C_1 \vee L_1 = X$, where $X^2 = 0$
- if circle and line have no intersection, $C_1 \vee L_1 = X$, where $X^2 \neq 0$

Equally, if $C_1 \vee L_1 = X$ and $X^2 = 0$, then the intersecting point is represented by X and if the meet gives zero, the two intersections can be found by intersecting the plane of the circle with the line. It is also important to mention that, in the case where $C_1 \vee L_1 = X$ and $X^2 \neq 0$, which means that no intersection exist, the sign of X^2 tell us whether the line passes through the circle ($X^2 < 0$) or does not pass through the circle ($X^2 > 0$).

Intersecting planes with planes, circles and lines

This section studies the intersection between planes and planes, planes and circles, and planes and lines, Firstly, the plane to plane intersection will be studied. Consider two planes Φ_1 and Φ_2 . The meet between two planes gives:

$$L = \Phi_1 \vee \Phi_2 = [\langle \Phi_1 \Phi_2 \rangle_{2n-r-s}]^* \quad (2.90)$$

where $2n - r - s = 10 - 4 - 4 = 2$. Therefore, the dual object has grade $5 - 2 = 3$, and represents a line. The sign of L^2 indicates whether the planes intersect. If $L^2 > 0$, then the planes intersect in the line L . On the other hand, when $L^2 = 0$, the intersection does not exist, which means that the two planes are parallel.

The intersection of a plane Φ_1 and a circle C_1 , when this exists, is a pair of points, or a single point and it is formulated as:

$$B = \Phi_1 \vee C_1 = [\langle \Phi_1 C_1 \rangle_{2n-r-s}]^* \quad (2.91)$$

where $2n - r - s = 10 - 4 - 3 = 3$, and so the dual object has grade 2. Looking at the sign of the resulting 2-blade, B , it is possible to elicit if the intersection exists and whether the intersection is a single point or a pair of points. Therefore, if $B^2 > 0$, then the meet between Φ_1 and C_1 gives two points. Given the bivector, B , the two points of the intersection can be extracted via the projectors given in equation 2.65. In the case where $B^2 = 0$, the intersection is a single point, X . It is trivial to find the representation of that point using the formula $X = BnB$. Finally, when $B^2 < 0$ holds, the intersection between the plane and the circle does not exist.

Replacing the circle C_1 with a line L_1 , the meet will still give us a 2-blade, B . However, a line and a plane intersect at most in one single position. Thus, the bivector will be of the form $B = X \wedge n$, where X is the representation of the point of the intersection. If $B^2 > 0$ the line and plane intersect in a point, if $B^2 = 0$ the line and plane are parallel and never intersect and if $B = 0$ the line lies in the plane.

Intersecting spheres with circles or lines

The intersection of a sphere, Σ_1 (4-blade), with a circle, C_1 (3-blade) or a single line, is the subject of this section. The intersection, where it exists, could be a single point or a pair of points (or the circle C_1 for the case where the circle is exactly on the outline of the sphere).

According to the meet formulation, the intersection of a sphere, Σ_1 , with a circle, C_1 can be expressed as:

$$B = \Sigma_1 \vee C_1 = [\langle \Sigma_1 C_1 \rangle_{2n-r-s}]^* \quad (2.92)$$

$2n - r - s = 10 - 4 - 3 = 3$, hence, the dual quantity will have grade $5 - 3 = 2$, which is a bivector, and represents the 2 intersecting points. Once again, the sign of the resulting squared 2-blade, B^2 , gives us information on whether the intersection exist and if the sphere and circle intersect in a single point or a pair of points. In the case of two intersections the points can be extracted from B using the projectors, as before, and in the case of tangency, the single point of contact is obtained by taking BnB .

Similarly, substituting the circle C_1 with a line L_1 and intersecting with a sphere Σ_1 , the meet again returns a 2-vector whose square denotes whether there are two, one or no intersection. The intersection points can be obtained easily in the same way as in the circle case.

Intersecting spheres with spheres or planes

This paragraph deals with the intersection of two spheres Σ_1 and Σ_2 or the intersection of a sphere Σ_1 and a plane Φ_1 . This intersection, where it exists, is a circle (or a single point).

Firstly, we will consider the intersection between two spheres. Spheres do not intersect if the distance from their centres is less than the sum of their radii $d > (\rho_0 + \rho_1)$. Also, if $d < |\rho_1 - \rho_2|$, one of the two spheres is completely contained in the other, hence no intersection exists. If $\rho_1 < \rho_2$, then the first sphere is contained in the second, otherwise the second sphere is contained in the first. In the case where $d = |\rho_1 - \rho_2| = 0$, then the two spheres are identical and the distance between them is trivially 0.

The two spheres are intersecting only when $|\rho_1 - \rho_2| \leq d \leq \rho_1 + \rho_2$. The intersection will be a circle (or a point, if the two spheres merely touch one another) with normal plane Φ . This circle can be calculated using the formula for the meet ([22]):

$$C = \Sigma_1 \vee \Sigma_2 = [\langle \Sigma_1 \Sigma_2 \rangle_{2n-r-s}]^* \quad (2.93)$$

$2n - r - s = 2 \cdot 5 - 4 - 4 = 2$, so that the dual quantity will have grade $5 - 2 = 3$, which is a trivector, and generally represents the circle of intersection. The value of C can tell us whether the result is a circle ($C^2 > 0$), a single point ($C^2 = 0$)

or there is no intersection ($C^2 < 0$). In the case where ($C^2 > 0$), the centre and radius of the circle can be extracted according to section 2.2.5. Similarly, using the same extracting formula from C for the case where ($C^2 = 0$), we will find that the circle has zero radius and its centre will be the point of tangency of the two spheres. In the same way, an attempt to extract the radius and the centre for the case where ($C^2 < 0$) leads to imaginary values for radius and centre (because no intersection exist) which lies on the shortest line joining the surface of the spheres (i.e. that joining the centres). If the two spheres have the same radii, it is the midway point on this line.

In the same way, instead of having a second sphere, Σ_2 , we can have an intersection with a plane, Φ_1 . The result of the meet between sphere and plane will also be a trivector, C ,

$$C = \Sigma_1 \vee \Phi_1 = [\langle \Sigma_1 \Phi_2 \rangle_2]^* \quad (2.94)$$

and the sign of the square of this trivector, C^2 indicates whether the two objects are tangent, intersect in a circle or do not intersect at all.

2.3. Conclusions

This chapter introduced Geometric Algebra and examined how the Conformal Model can be used to represent geometric primitives such as points pairs, lines, circles, planes and spheres. Also it is studied how rotors can be applied to those objects, within the Conformal Model. The model described in this chapter will be the basis on which this report will be based on.

3

Obtaining 3D marker positions

Motion capture hardware, such as Phasespace [25], CodaMotion [26] and Vicon [27], are 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 mobile gait labs; or for visualisation of virtual characters for films and computer games [6]. Throughout this work, a marked motion capture 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 mark-to-limb association, which are 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.

3.1. 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 Phases-

pace [25]. 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. This box must initially be wired to the main control unit for the cameras in order to synchronise the LED modulation with the capture of the cameras. After a short period it can be disconnected to allow wireless operation for a reasonably long period of time.

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.

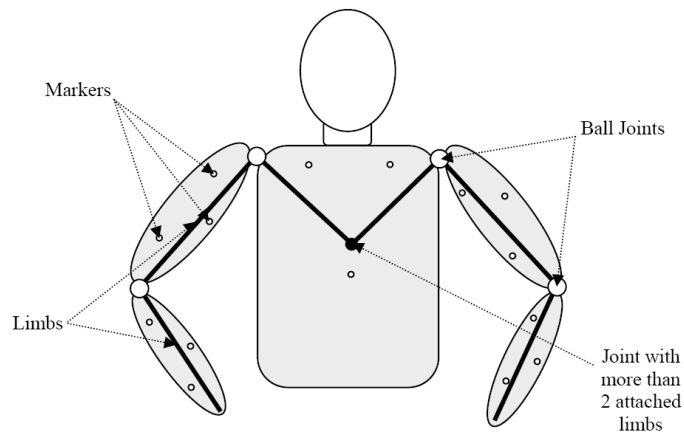


Figure 3.1.: Marker-to-limb association example with three markers on each limb. This example shows the upper body with the marker-to-limb association.

3.2. 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.

The concept analysed above was implemented in MATLAB 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 3.1.

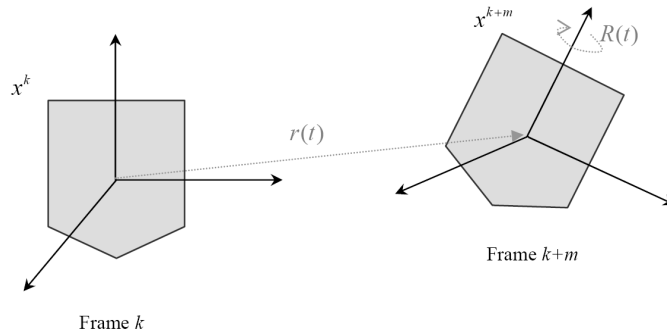


Figure 3.2.: Description of the rigid body. The vector $r(t)$ specifies the position of the centre of mass, relative to the origin. The rotor $R(t)$ defines the orientation of the body, relative to a fixed copy placed at the origin. The x^k is a vector in a reference body (in this example in frame k), and x^{k+m} is a vector in space of the equivalent point of the moving body (in this example after m frames).

3.3. 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. More specifically, two vector variables can describe such a system, one for position of the centre of mass, and one for the vector from the centre of mass to a point in the body. The final body position can therefore be expressed in terms of a rotation and translation from a fixed “reference” body on to the body in space (figure 3.2). 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:

$$x^{k+m} = Rx^kR^T + r(t) \quad (3.1)$$

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 $R(t)$.

4

Real Time Joint Localisation

4.1. 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 localisation 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.

4.2. Related Work

Several papers have focused on methods for real-time localisation of the centre of rotation. These methods can be separated into two major groups. In the first method, also known as the *sphere fitting* method, 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* method, 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 [28], 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 localisation 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. However, this algorithm requires a series of weights to be set such as a perceived accuracy of a given marker and a spatial reweighting of data points via a voxel grid. This method also has the disadvantages that the non-linear nature of the solver makes it susceptible to problems with local minima and the inaccuracy of the heuristics used to set the weight parameters.

Halvorsen et al., in [8], assign a closed form solution using the geometric properties of the sphere. They used the fact that the perpendicular bisectors of chords on 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 thus the accuracy and effectiveness of the algorithm.

In [29], 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

$$((C - x_m^k)^2 - (r_m)^2)^2 \quad (4.1)$$

Cereatti et al, [9], shows that, the cost function used in [29] corresponds to the minimised cost function used in [8].

An alternative approach provided by Halvorsen, [10], gives a Bayesian analysis of the algorithm of [29], 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 direction¹. Hence, the cost function of [29] can be written as:

$$\begin{aligned}
 S &= \sum_k \sum_m ((e_m^k + r_m)^2 - (r_m)^2) \\
 &= \sum_k \sum_m ((e_m^k)^2 + 2r_m e_m^k)
 \end{aligned}
 \tag{4.2}$$

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 [29] was recently used in Cerveri et al. ([30], [31]) who attempted to find an optimal fit for a hand skeleton.

Transformation

The transformation 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 implemented in [12]. This technique makes it possible to determine limb lengths, joint locations, and sensor placement for a human subject without external measurements, but 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 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 ([29], [32]) and no user feedback to set marker weights, as in [7], are needed. The method is closed form, thus enabling real-time implementation.

4.3. 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 different

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

clustering techniques. The next stage is the localisation of the joints in each frame. However, before the localisation of the joints, another problem occurs. we also have the problem of how to express the rotation of two limbs between frames. We 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 [33].

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

$$R = \arg \max \sum_{i=1}^n (Rx_i \tilde{R}) \cdot y_i \quad (4.3)$$

In order to avoid introducing quaternion terminology and methods, we can express the solution in terms of Geometric Algebra. To clarify the following equations, the marker index i will temporally be suppressed. Let:

$$R = a + B_1 e_{23} + B_2 e_{13} + B_3 e_{12} \quad (4.4)$$

$$x_i = x_1 e_1 + x_2 e_2 + x_3 e_3 \quad y_i = y_1 e_1 + y_2 e_2 + y_3 e_3 \quad (4.5)$$

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 [15]. Each of these encodes a distinct plane, and there are 3 of them to match the 3 independent planes in the 3-D space.

Then:

$$\begin{aligned} (Rx_i \tilde{R}) \cdot y_i &= a^2(x \cdot y) + 2aB_1(x_3y_2 - x_2y_3) + 2aB_2(x_3y_1 - x_1y_3) \\ &+ 2aB_3(2x_2y_1 - 2x_1y_2) + B_1^2(x_1y_1 - x_2y_2 - x_3y_3) \\ &+ 2B_1B_2(-x_2y_1 - x_1y_2) + 2B_1B_3(x_3y_1 + x_1y_3) \\ &+ B_2^2(-x_1y_1 + x_2y_2 - x_3y_3) + 2B_2B_3(-x_3y_2 + x_2y_3) \\ &+ B_3^2(-x_1y_1 - x_2y_2 + x_3y_3) \end{aligned}$$

This can be expressed as follows:

$$R = \arg \max \sum_{i=1}^n (R_v^T N R_v) \quad (4.6)$$

where:

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

$$N = \begin{bmatrix} x_1y_1 + x_2y_2 + x_3y_3 & x_3y_2 - x_2y_3 & x_3y_1 - x_1y_3 & x_2y_1 - x_1y_2 \\ x_3y_2 - x_2y_3 & x_1y_1 - x_2y_2 - x_3y_3 & -(x_2y_1 + x_1y_2) & x_3y_1 + x_1y_3 \\ x_3y_1 - x_1y_3 & -(x_2y_1 + x_1y_2) & -x_1y_1 + x_2y_2 - x_3y_3 & x_3y_2 - x_2y_3 \\ x_2y_1 - x_1y_2 & x_3y_1 + x_1y_3 & x_3y_2 + x_2y_3 & -x_1y_1 - x_2y_2 + x_3y_3 \end{bmatrix}$$

In terms of limb motion we note that x_1, x_2, x_3 and y_1, y_2, y_3 correspond to values given below, with x_i^k, y_i^k, c_x^k and c_y^k defined in figure 4.1.

$$\begin{aligned} x_1 &= x_1^k - c_x^k & y_1 &= y_1^k - c_y^k \\ x_2 &= x_2^k - c_x^k & y_2 &= y_2^k - c_y^k \\ x_3 &= x_3^k - c_x^k & y_3 &= y_3^k - c_y^k \end{aligned} \quad (4.7)$$

$$c_x^k = \frac{x_1^k + x_2^k + x_3^k}{3} \quad c_y^k = \frac{y_1^k + y_2^k + y_3^k}{3} \quad (4.8)$$

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

$$R = \arg \max \left(\frac{R_v^T \sum_{i=1}^n (N) R_v}{R_v^T R_v} \right) \quad (4.9)$$

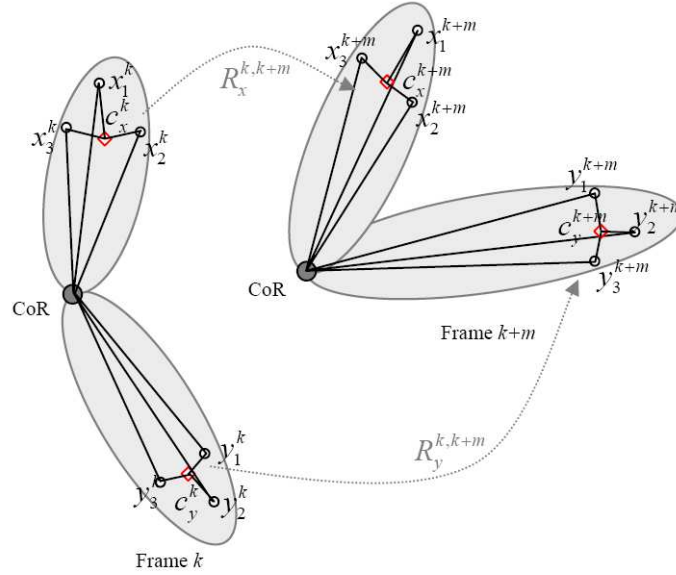


Figure 4.1.: The motion of two limbs with a time-difference of m frames.

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

of $\sum_{i=1}^n (N)$, thus providing 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. This procedure reduces the computational cost from $O(m^3)$ to $O(m)$.

4.4. 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 centre of rotation.

4.4.1. Cost function

The method used to estimate the centres of rotation is a variation of the least squares solution proposed in [34]. The main principles are illustrated below.

Suppose we place markers on two joint limbs (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 joints 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:

$$x_i^k = C^k + R_x^k a_x^i \tilde{R}_x^k \quad y_j^k = C^k + R_y^k a_y^j \tilde{R}_y^k \quad (4.10)$$

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

$$b_{ij}^k = x_i^k - y_j^k = R_x^k a_x^i \tilde{R}_x^k - R_y^k a_y^j \tilde{R}_y^k \quad (4.11)$$

Figure 4.2 presents a typical marker placement with the relevant markers with the b_{ij}^k , a_x^i and a_y^j as shown.

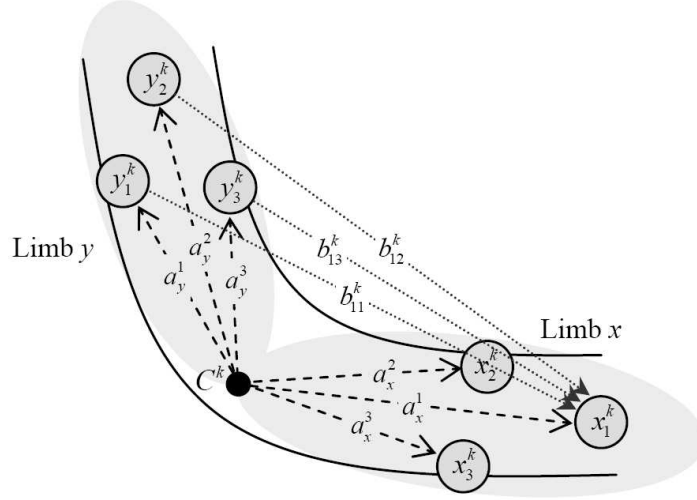


Figure 4.2.: A typical marker placement with relevant markers shown.

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.

$$S = \sum_{k=1}^m \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \left[b_{ij}^k - \left(R_x^k a_x^i \tilde{R}_x^k - R_y^k a_y^j \tilde{R}_y^k \right) \right]^2 \quad (4.12)$$

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.

4.4.2. 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 $\partial a(a \cdot b) = b$.

$$\partial_{a_x^i} S = 2 \sum_{k=1}^m \sum_{j=1}^{n_y} \left[-\tilde{R}_x^k b_{ij}^k R_x^k + a_x^i - \tilde{R}_x^k R_y^k a_y^j \tilde{R}_y^k R_x^k \right] \quad (4.13)$$

$$\partial_{a_y^j} S = 2 \sum_{k=1}^m \sum_{i=1}^{n_x} \left[-\tilde{R}_y^k b_{ij}^k R_y^k + a_y^j - \tilde{R}_y^k R_x^k a_x^i \tilde{R}_x^k R_y^k \right] \quad (4.14)$$

In order to get the minimum of the cost function, we require the partial derivatives to be equal to 0. Thus, from equation 4.13 we obtain:

$$\begin{aligned}
\sum_{k=1}^m \sum_{j=1}^{n_y} a_x^i &= \sum_{k=1}^m \sum_{j=1}^{n_y} \left\{ \tilde{R}_x^k b_{ij}^k R_x^k + \tilde{R}_x^k R_y^k a_y^j \tilde{R}_y^k R_x^k \right\} \\
mn_y a_x^i &= \sum_{k=1}^m \sum_{j=1}^{n_y} \left\{ \tilde{R}_x^k b_{ij}^k R_x^k + \tilde{R}_x^k R_y^k a_y^j \tilde{R}_y^k R_x^k \right\} \\
a_x^i &= \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k b_i^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k
\end{aligned} \tag{4.15}$$

$$\begin{aligned}
\bar{a}_x &= \frac{1}{n_x} \sum_{i=1}^{n_x} \left[\frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k b_i^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k \right] \\
\bar{a}_x &= \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k
\end{aligned} \tag{4.16}$$

where

$$\bar{a}_x = \frac{1}{n_x} \sum_{i=1}^{n_x} a_x^i \qquad \bar{a}_y = \frac{1}{n_y} \sum_{j=1}^{n_y} a_y^j \tag{4.17}$$

$$b_i^k = \frac{1}{n_y} \sum_{j=1}^{n_y} b_{ij}^k \qquad b_j^k = \frac{1}{n_x} \sum_{i=1}^{n_x} b_{ij}^k \tag{4.18}$$

and

$$\bar{b}^k = \frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} b_{ij}^k \tag{4.19}$$

Similarly, from equation 4.14:

$$\begin{aligned}
\sum_{k=1}^m \sum_{i=1}^{n_x} a_y^j &= \sum_{k=1}^m \sum_{i=1}^{n_x} \left\{ -\tilde{R}_y^k b_{ij}^k R_y^k + \tilde{R}_y^k R_x^k a_x^i \tilde{R}_x^k R_y^k \right\} \\
mn_x a_y^j &= \sum_{k=1}^m \sum_{i=1}^{n_x} \left\{ -\tilde{R}_y^k b_{ij}^k R_y^k + \tilde{R}_y^k R_x^k a_x^i \tilde{R}_x^k R_y^k \right\} \\
a_y^j &= \frac{1}{m} \sum_{k=1}^m -\tilde{R}_y^k b_j^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k
\end{aligned} \tag{4.20}$$

$$\begin{aligned}
\bar{a}_y &= \frac{1}{n_y} \sum_{i=1}^{n_y} \left[\frac{1}{m} \sum_{k=1}^m -\tilde{R}_y^k b_j^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k \right] \\
\bar{a}_y &= \frac{1}{m} \sum_{k=1}^m -\tilde{R}_y^k \bar{b}^k R_y^k + \frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k
\end{aligned} \tag{4.21}$$

Substituting equation 4.21 into equation 4.16 we obtain:

$$A_1 \bar{a}_x = d_1 \quad (4.22)$$

where:

$$\begin{aligned} A_1 &= \left[I - \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_x^k R_y^k \left(\sum_{k'=1}^m \tilde{R}_y^{k'} R_x^{k'} \tilde{R}_x^{k'} R_y^{k'} \right) \tilde{R}_y^k R_x^k \right\} \right] \\ d_1 &= \frac{1}{m} \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k - \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_x^k R_y^k \left(\sum_{k'=1}^m \tilde{R}_y^{k'} \bar{b}^{k'} R_y^{k'} \right) \tilde{R}_y^k R_x^k \right\} \end{aligned} \quad (4.23)$$

Similarly, substituting the equation 4.16 into equation 4.20 we obtain:

$$A_2 \bar{a}_y = d_2 \quad (4.24)$$

where:

$$\begin{aligned} A_2 &= \left[I - \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_y^k R_x^k \left(\sum_{k'=1}^m \tilde{R}_x^{k'} R_y^{k'} \tilde{R}_y^{k'} R_x^{k'} \right) \tilde{R}_x^k R_y^k \right\} \right] \\ d_2 &= -\frac{1}{m} \sum_{k=1}^m \tilde{R}_y^k \bar{b}^k R_y^k + \frac{1}{m^2} \sum_{k=1}^m \left\{ \tilde{R}_y^k R_x^k \left(\sum_{k'=1}^m \tilde{R}_x^{k'} \bar{b}^{k'} R_x^{k'} \right) \tilde{R}_x^k R_y^k \right\} \end{aligned} \quad (4.25)$$

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

$$\bar{a}_x = (A_1^T A_1)^{-1} A_1^T d_1 \quad \bar{a}_y = (A_2^T A_2)^{-1} A_2^T d_2 \quad (4.26)$$

Since $R_x^k a_x^i \tilde{R}_x^k = x_i^k - C^k$ (equation 4.10), then:

$$\begin{aligned} R_x^k \bar{a}_x \tilde{R}_x^k &= \frac{1}{n_x} \sum_{i=1}^{n_x} x_i^k - C^k \\ \Rightarrow C^k &= \frac{1}{n_x} \sum_{i=1}^{n_x} x_i^k - R_x^k a_x^i \tilde{R}_x^k \end{aligned} \quad (4.27)$$

Similarly,

$$\begin{aligned} R_y^k \bar{a}_y \tilde{R}_y^k &= \frac{1}{n_y} \sum_{j=1}^{n_y} y_j^k - C^k \\ \Rightarrow C^k &= \frac{1}{n_y} \sum_{j=1}^{n_y} y_j^k - R_y^k a_y^j \tilde{R}_y^k \end{aligned} \quad (4.28)$$

Hence,

$$C^k = \frac{1}{2n_x} \sum_{i=1}^{n_x} x_i^k + \frac{1}{2n_y} \sum_{j=1}^{n_y} y_j^k - \frac{1}{2} \left(R_x^k \bar{a}_x \tilde{R}_x^k + R_y^k \bar{a}_y \tilde{R}_y^k \right) \quad (4.29)$$

The above solution is a matrix-based approach. We observed that equations 4.16 and 4.21 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:

$$\begin{aligned} m a_x^i &= \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k + \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k \\ m a_x^i - \sum_{k=1}^m \tilde{R}_x^k R_y^k \bar{a}_y \tilde{R}_y^k R_x^k &= \sum_{k=1}^m \tilde{R}_x^k \bar{b}^k R_x^k \end{aligned} \quad (4.30)$$

Similarly,

$$\begin{aligned} m a_y^j &= \sum_{k=1}^m \tilde{R}_y^k \bar{b}^k R_y^k + \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k \\ m a_y^j - \sum_{k=1}^m \tilde{R}_y^k R_x^k \bar{a}_x \tilde{R}_x^k R_y^k &= \sum_{k=1}^m \tilde{R}_y^k \bar{b}^k R_y^k \end{aligned} \quad (4.31)$$

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

$$M a = b \quad (4.32)$$

where

$$M = \begin{bmatrix} mI_3 & M_x \\ M_y & mI_3 \end{bmatrix} \quad (4.33)$$

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, the size of the matrix requiring inversion can be reduced to a constant 6×6 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 4.33 can be achieved using the following transformation:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix} \quad (4.34)$$

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 3×3 matrix. Thus we exchange a single 6×6 matrix inverse for a 3×3 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 4.16 and 4.21, 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 4.3 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 [35]. 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 over time). Hence, using eq. 4.10 the CoR can be estimated using information from the limb segment which has all its markers visible. Thus,

$$\hat{C}^k = w_i^k - R_w^k a_w^i \tilde{R}_w^k \quad (4.35)$$

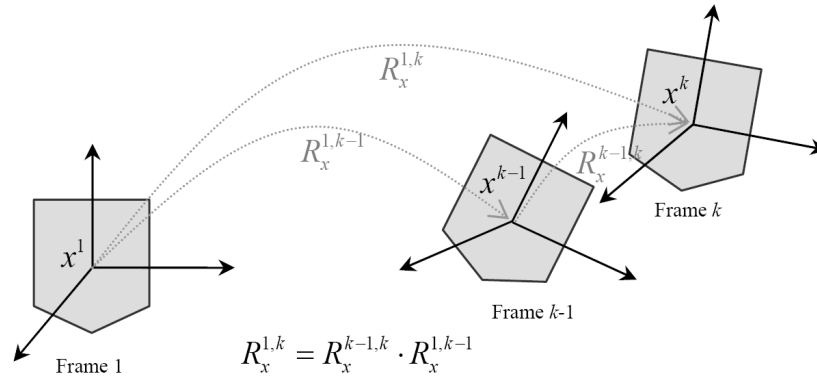


Figure 4.3.: The rotors during frames.

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.

5

Estimating the occluded markers

The Phasespace [25] 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 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 have 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.

5.1. 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 [36, 37, 38, 39, 40]; 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, [41], 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 [42] used an extended Kalman filter (EKF) to predict the missing markers using previously available marker information while Welch et al. in [43] 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 [44] and [45], 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. [46] 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, [35], also present an automatic method to identify indistinguishable markers based on cliques¹. However, this requires an off-line procedure in order to determine marker cliques and parameters of the skeletal structure.

In [47], 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, [48], presents 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 [49, 50], 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 relative 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 equation ??). Therefore, a method based on the prediction of the occluded markers will be more practicable.

5.2. Kalman Filter

The Kalman filter [51] is a commonly used method for tracking in many different areas, such as autonomous or assisted navigation, interactive computer graphics and motion prediction. The simplicity and the robust nature of the filter

¹Markers in a clique have constant distances between each other

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 (5.1), where the state \mathbf{x}_k at frame k is obtained from the state at frame $k - 1$;

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (5.1)$$

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

$$\mathbf{Z}_k = H\mathbf{x}_k + \mathbf{v}_k \quad (5.2)$$

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

$$\mathbf{y}_k = A\hat{\mathbf{x}}_{k-1} + B\mathbf{u}_{k-1} \quad E_k = AP_{k-1}A^T + Q \quad (5.3)$$

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:

$$K_k = E_k H^T (H E_k H^T + R)^{-1} \quad (5.4)$$

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

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + K_k (\mathbf{Z}_k - H\mathbf{y}_k) \quad (5.5)$$

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;

$$P_k = (I - K_k H) E_k \quad (5.6)$$

In this work 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:

$$\mathbf{y}_k = \mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1} dk \quad (5.7)$$

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

$$\begin{bmatrix} \mathbf{x}_k \\ \dot{\mathbf{x}}_k \end{bmatrix} = \begin{bmatrix} 1 & dk \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \dot{\mathbf{x}}_{k-1} \end{bmatrix} + B\mathbf{u}_{k-1} \quad (5.8)$$

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 $A = \begin{bmatrix} I_3 & I_3 \\ 0 & I_3 \end{bmatrix}$.

More information on the Kalman filter parameters and an alternative approach, using only the position of previous frames instead of the velocity, are presented in Appendices A.2 and A.1, respectively.

5.3. 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.

5.3.1. All markers are visible on a given limb

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

$$\mathbf{Z}_k = H\mathbf{x}_k + \mathbf{v}_k \quad (5.9)$$

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).

5.3.2. One missing marker on a limb segment

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

$$\mathbf{Z}_k = H\hat{\mathbf{x}}_1^k + \mathbf{v}_k \quad (5.10)$$

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 5.1):

$$\mathbf{D}_{i,j}^{k-1} = \mathbf{x}_j^{k-1} - \mathbf{x}_i^{k-1} \quad (5.11)$$

Thereafter, these vectors are rotated as

$$\hat{\mathbf{D}}_{i,j}^k = R^{k-2,k-1}\mathbf{D}_{i,j}^{k-1}\tilde{R}^{k-2,k-1} \quad (5.12)$$

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;

$$\tilde{\mathbf{x}}_1^k = \frac{(\mathbf{x}_2^k - \hat{\mathbf{D}}_{1,2}^k) + (\mathbf{x}_3^k - \hat{\mathbf{D}}_{1,3}^k)}{2} \quad (5.13)$$

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 of this problem is given in Appendix C. Figure 5.1 illustrates this process.

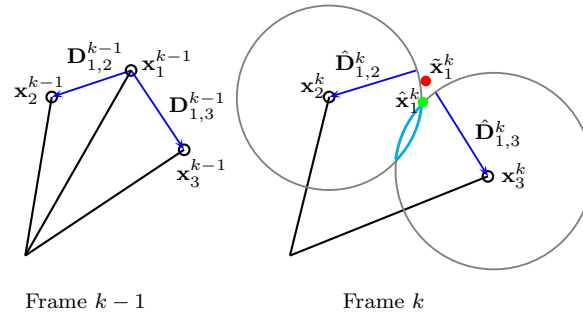


Figure 5.1.: The observation vector in the case of 2 visible markers. The red dot, $\tilde{\mathbf{x}}_1^k$, represents the average value as given in equation 5.13. The green dot, $\hat{\mathbf{x}}_1^k$, is the point on the intersection of the 2 spheres which is closest to $\tilde{\mathbf{x}}_1^k$.

5.3.3. Two missing markers on a limb segment

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

$$\mathbf{Z}_k = H\hat{\mathbf{x}}_j^k + \mathbf{v}_k \quad (5.14)$$

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:

$$\hat{\mathbf{x}}_j^k = \mathbf{x}_2^k - \hat{\mathbf{D}}_{j,2}^k \quad (5.15)$$

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 5.2 demonstrates the above.

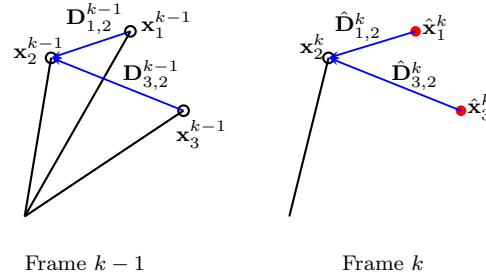


Figure 5.2.: The observation vector in the case of only one visible marker. The red dots, $\hat{\mathbf{x}}_j^k$, where $j = \{1, 3\}$ represent the estimated position of the missing marker m_j as given in equation 5.15.

5.3.4. 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 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 4.4.2, equation 4.35. In that case the observation vector of the Kalman filter is updated as:

$$\mathbf{Z}_k = H\hat{\mathbf{x}}_j^k + \mathbf{v}_k \quad (5.16)$$

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;

$$\hat{\mathbf{x}}_j^k = \hat{\mathbf{C}}_k + \hat{\mathbf{D}}_{j,c}^k \quad (5.17)$$

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 5.3 illustrates the above.

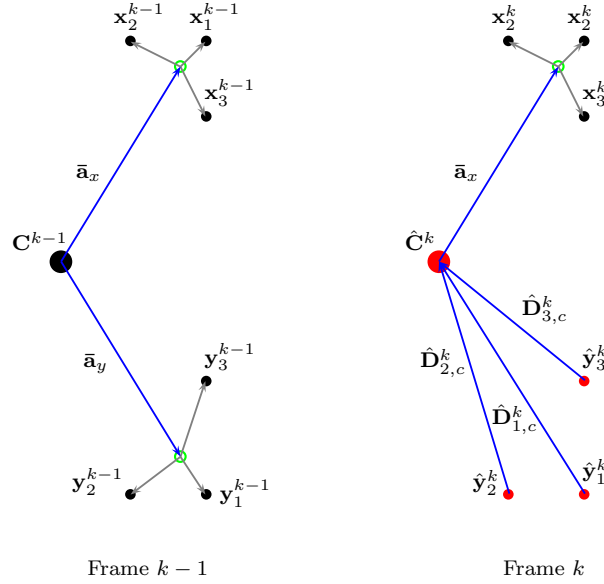


Figure 5.3.: The estimation procedure when all markers on a single limb segment are occluded. The red dots represents the estimated position of the CoR, $\hat{\mathbf{C}}^k = \mathbf{x}_i^k - R_x^k \bar{\mathbf{a}}_x \tilde{R}_x^k$ where $i = \{1, 2, 3\}$, and the estimated marker positions of the limb segment y , $\hat{\mathbf{y}}_j^k = \hat{\mathbf{C}}^k + \hat{\mathbf{D}}_{j,c}^k$, 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

$$\mathbf{Z}_k = H \hat{\mathbf{x}}^k + \mathbf{v}_k \quad (5.18)$$

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:

$$\mathbf{Z}_t = H \mathbf{x}_1^{t-1} + \mathbf{U}_t \quad (5.19)$$

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 CoR is calculated with a bigger error.

5.3.5. 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 5.4. 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.

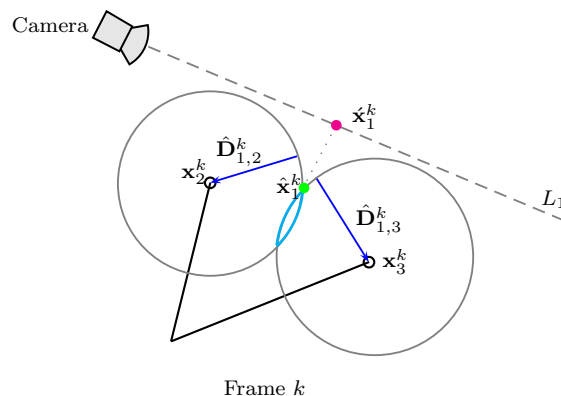


Figure 5.4.: The observation vector in the case of 2 visible markers and one marker visible only by one camera. The magenta dot, $\hat{\mathbf{x}}_1^k$, is now used for the calculation of the observation vector, $\mathbf{Z}_k = H\hat{\mathbf{x}}^k + \mathbf{v}_k$.

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 Σ_2 where Σ_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 Appendices D and E respectively. 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.

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 [25]. The algorithms were implemented in MATLAB and run on a Pentium IV PC. The proposed system can process up to 350 *frames 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, the results of our experiments will be presented. Using real data with occlusions generated by artificial deletion, we calculated the error of the proposed methodologies under real conditions; error is equal to the average

distance between the true position and the estimated position of the marker and the CoR respectively. The error varies between different instances of marker occlusion. As more markers 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 30 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 and $\bar{\mathbf{a}}_w$ is 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 remains constant during time). 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. The missing marker introduces more noise in the system than calculating the CoR using information only from one limb segment. However, $\bar{\mathbf{a}}_w$ should be updated using the predicted positions of the markers. Figure 6.1 presents the localisation of the CoR for the case where a marker on each limb segment is missing. Red indicates the true position of markers and limbs, and blue represents the predicted positions.

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 missing markers are 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

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 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 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 5.3.4. 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 markers 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 positions value, 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 period 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 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 6.2 shows how the marker position errors using the Kalman filter for both constant velocity and constant rotation models compare for different number 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 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 5.3.5, 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.

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. 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

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. 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

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.

Figures 6.4 and 6.5 show examples of the output of the proposed algorithms on real data. Figure 6.4 shows the lower body of a female as she runs and figure 6.5 shows the upper body of a male as he dances. These illustrative examples show that the proposed method can estimate the position of the occluded markers and locate the joint parameters in real-time with high accuracy.

6.3. Discussion

The work outlined in this Chapter investigates reconstruction of marker position for multiple cases of occlusion. The 3*d* 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 4.35, 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 algorithm, 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.

Experiments demonstrate that the method presented effectively recovers, in real-time, a good estimate of the true positions of the missing markers and reconstructs the CoR location under multiple occlusions even if markers are not visible for extended periods of time. Part of these results will be published in [52] and [53].

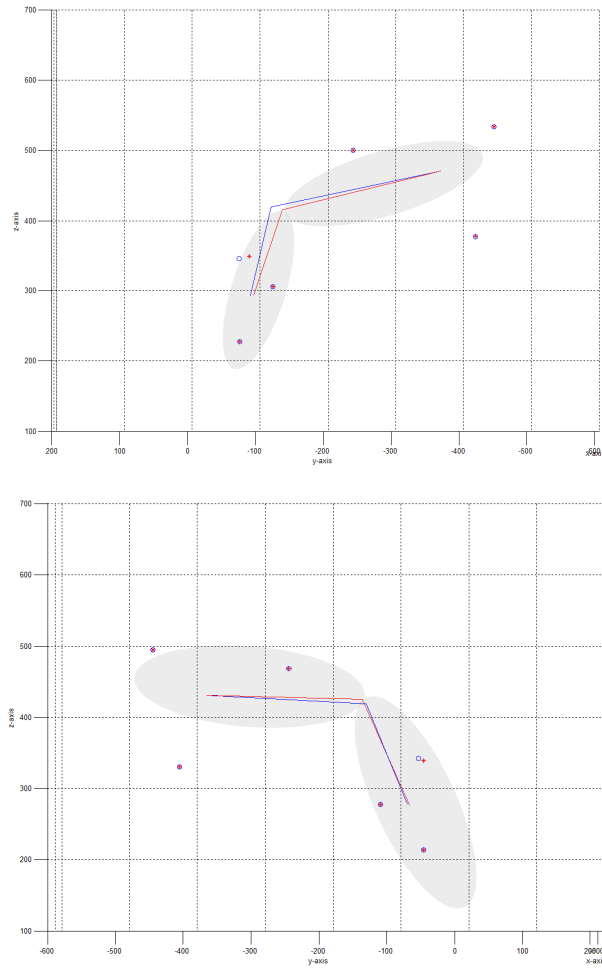


Figure 6.1.: Examples of localisation of the CoR, where two non-visible markers exist (1 on each limb). Red dots indicate the true position of markers where red lines represent the reconstructed limbs. Blue represents the predicted positions.

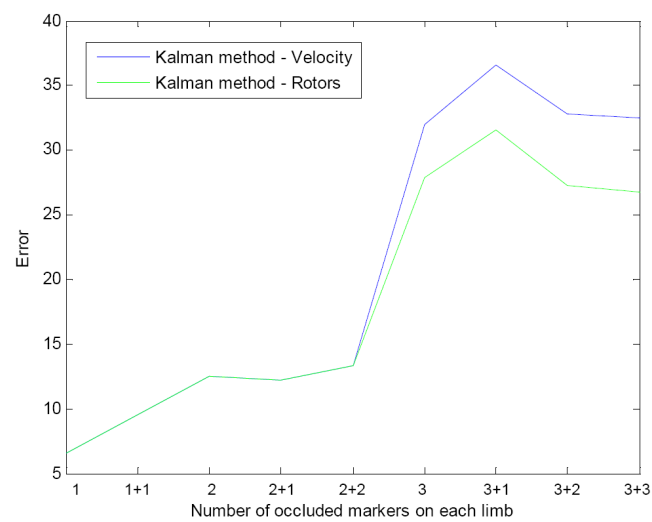


Figure 6.2.: The Kalman filter error using the *constant velocity* and the *rotors* approaches, against the number of occluded markers on each limb segment. For example, $2 + 1$ means 2 occluded markers on the first limb segment and 1 occluded marker on the second.

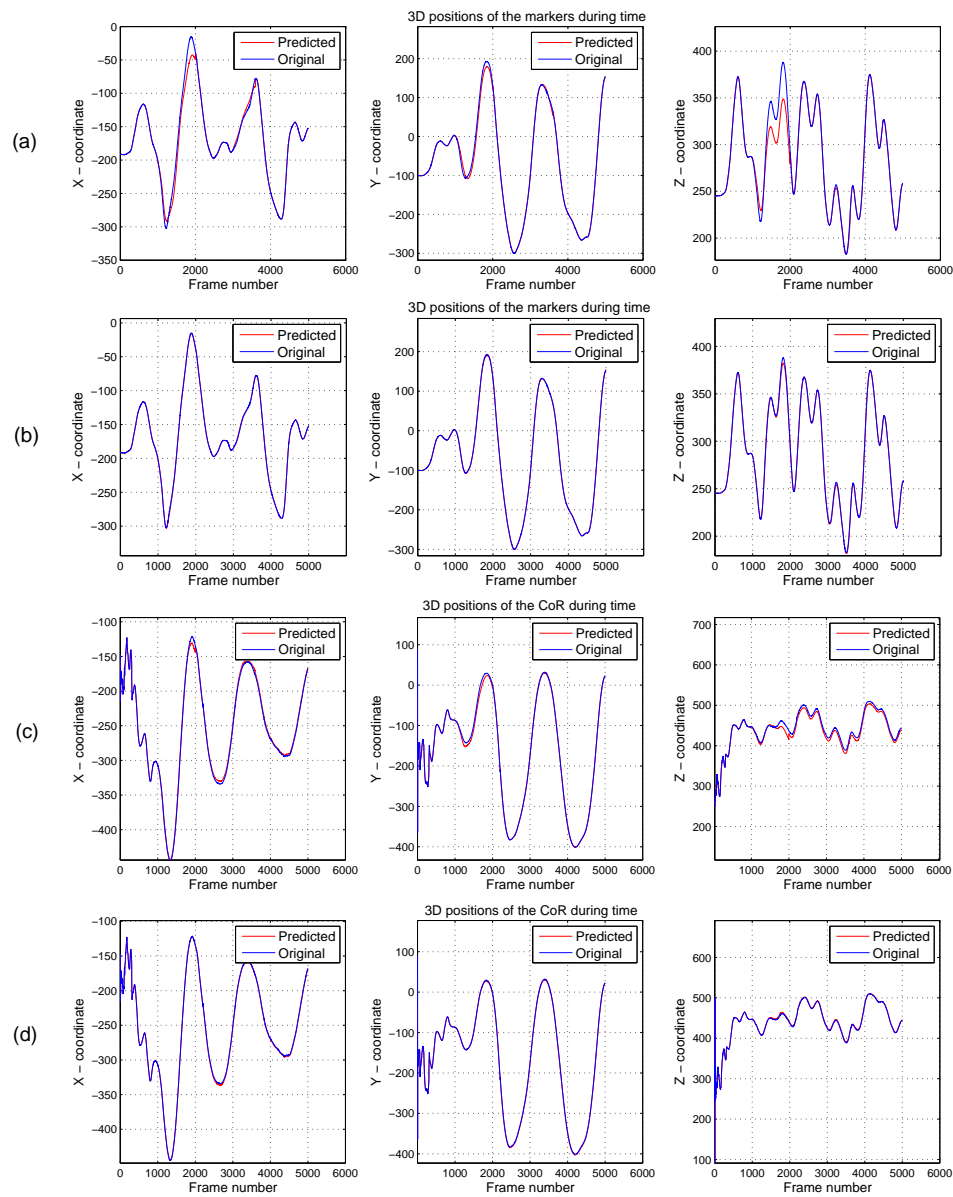


Figure 6.3.: An example of the 3D positions of predicted and the true coordinates of the *Markers* (a), (b) in the case of one missing marker on a limb segment and a missing marker visible by one camera respectively, and the *CoR* (c), (d) under the same conditions. The occlusion periods are between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

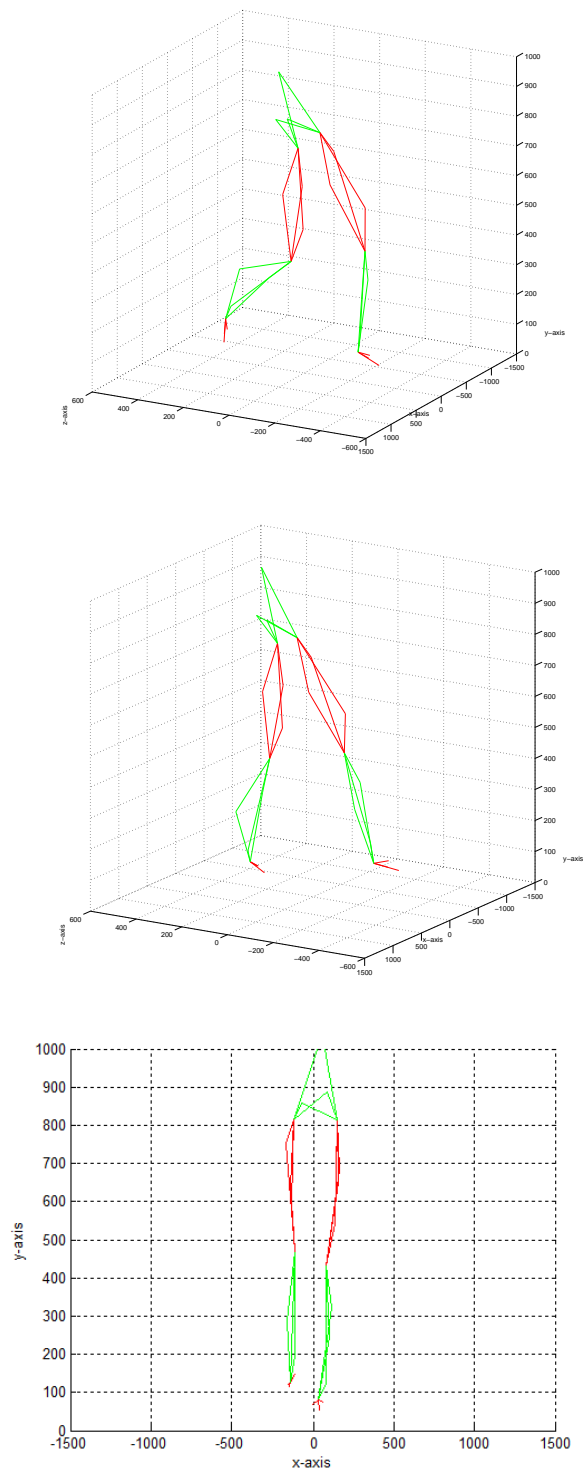


Figure 6.4.: Examples of implementation on real data (Lower body).

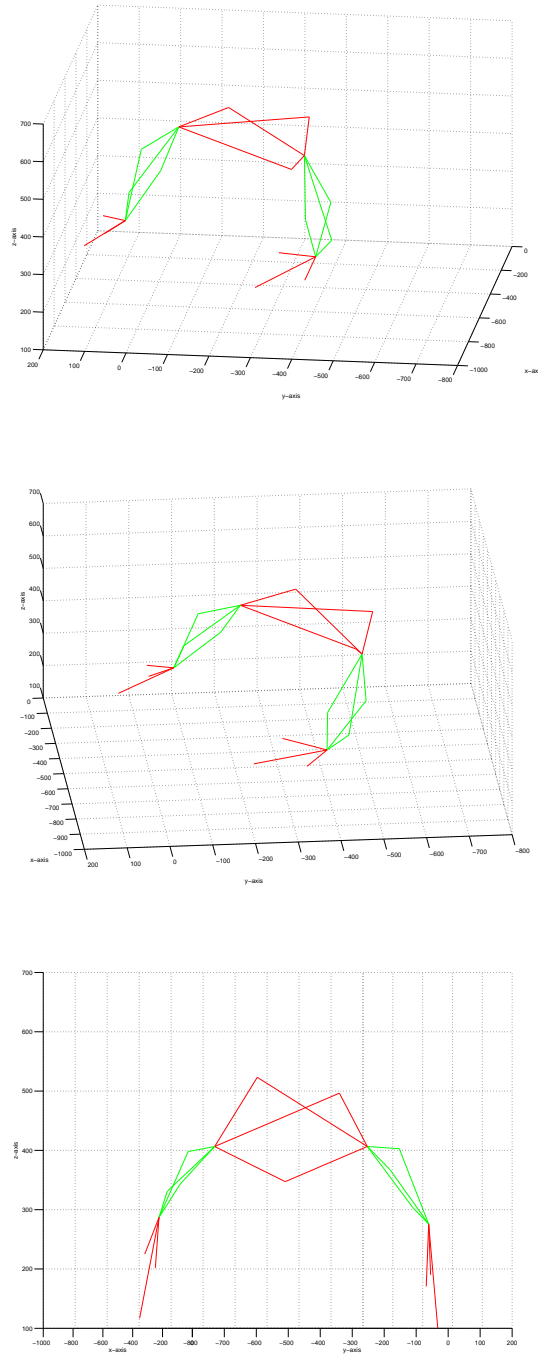


Figure 6.5.: Examples of implementation on real data (Arms).

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 gives a brief introduction to Geometric Algebra and introduces Conformal Geometric Algebra. Section two considers the problem of fitting skeletal models to marker-based optical motion capture data. Using the well know *Procrustes* formulation, it shows how to establish estimates of the relative orientation of a limb on 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.

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 the human motion. Also, a reliable model with which to predict the rotors expressing the rotation of each limb segment is needed.

APPENDICES



A

Kalman Filter

A.1. An alternative approach

Section 5.2 presents a Kalman filter that predicts the next position using a dynamic model. The equation which deals with that dynamic model and expresses the next possible position of the marker can be formulated as given below:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1}dk \quad (\text{A.1})$$

where \mathbf{x}_k corresponds to the position of the marker in frame k , \mathbf{x}_{k-1} is the position in frame $k - 1$ and $\dot{\mathbf{x}}_{k-1}$ is the velocity on the marker in frame $k - 1$. The assumption made here is that the acceleration remains constant over two consecutive frames.

Hence, equation 5.3, which expresses the predicted state, can be solved in a matrix model as given below:

$$\begin{bmatrix} \mathbf{x}_k \\ \dot{\mathbf{x}}_k \end{bmatrix} = \begin{bmatrix} 1 & dk \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \dot{\mathbf{x}}_{k-1} \end{bmatrix} + B\mathbf{u}_{k-1} \quad (\text{A.2})$$

In this section, an alternative method that uses a history model is presented, using information only from the previous frames, thus reducing the computational cost. Therefore, it will not be necessary to calculate the velocity of the marker $\dot{\mathbf{x}}_k$ in the current frame.

Assuming that $dk = 1$, we can write $\dot{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$. Hence, the 2^{nd} order dynamic model will be as given below:

$$\underbrace{\begin{bmatrix} \mathbf{x}_k \\ \dot{\mathbf{x}}_k \end{bmatrix}}_{\text{dynamics vector}} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix}}_{\text{history vector}} \quad (\text{A.3})$$

Substituting equation A.3 into equation A.2, and assuming that $B\mathbf{u}_{k-1} = 0$, we obtain:

$$\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-2} \end{bmatrix} \quad (\text{A.4})$$

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}^{-1}}_{B^{-1}} \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}}_B \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-2} \end{bmatrix} \quad (\text{A.5})$$

The inverse of the matrix B is equal to:

$$B^{-1} = \frac{B^T}{\det(B)}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}^{-1} = \frac{\begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}}{\det\left(\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}\right)} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \quad (\text{A.6})$$

Hence,

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-2} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-2} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-2} \end{bmatrix} \quad (\text{A.7})$$

Looking at equation A.7, we can see that the second row returns $\mathbf{x}_{k-1} = \mathbf{x}_{k-1}$, thus we can simplify the 2×2 matrix to an 1×2 matrix, as given below:

$$\begin{bmatrix} \mathbf{x}_k \end{bmatrix} = \begin{bmatrix} 2 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-2} \end{bmatrix} \quad (\text{A.8})$$

It is important here to remind ourselves that each marker position is constituted by x , y and z coordinates. Thus, the matrix A which updates the state can be expressed as:

$$A = \begin{bmatrix} 2I_3 & -I_3 \\ I_3 & 0 \end{bmatrix} \quad (\text{A.9})$$

The results of this approach, as expected, are the same as the method proposed in section 5.2. However, the processing time has not been improved since the Kalman filter cannot exploit the advantages offered from the 1×2 history matrix (see equation A.8). This happens because matrix A should be a square matrix. The 2×2 matrix (see equation A.9) is therefore used.

A.2. Kalman filter parameters

The Kalman filter parameters used for our experiments are given below:

The System noise is equal to:

$$Q = 1 \times I_6 \quad (\text{A.10})$$

The Measurement Noise has been estimated empirically and it is given by:

$$R = \begin{bmatrix} 0.2845 & 0.0455 & 0.0045 \\ 0.0455 & 0.0045 & 0.2845 \\ 0.0045 & 0.2845 & 0.0455 \end{bmatrix} \quad (\text{A.11})$$

The Observation Process is:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.12})$$

B

Finding the position of the occluded points using the rotors

This section presents an alternative way of estimating the positions of occluded markers in the current frame, using the rigid body information. In the method discussed in section 5.3, the vectors between the tracked marker \mathbf{x}_1^{k-1} and the two visible markers in frame $k-1$ have been calculated. Then, these vectors were added to the current position of the markers \mathbf{x}_2^t and \mathbf{x}_3^t , respectively. Thus, we had two different positions for \mathbf{x}_1^k . The final position of the marker, $\tilde{\mathbf{x}}_1^k$, is defined as the mid-point of these two positions. Now, instead of calculating the distances between each marker, we calculate the rotors $R_{1,2}^{k-1}$, $R_{1,3}^{k-1}$, $R_{2,3}^{k-1}$ expressing the rotation between each pair of markers. It is assumed that the rotation between each marker will vary slowly during time. Therefore, the rotors are applied to the current position of the two visible markers returning two different values for the current position of \mathbf{x}_1^k . Hence, $\tilde{\mathbf{x}}_1^k$ is defined as the mid-point of these two positions.

$$\tilde{\mathbf{x}}_1^k = \frac{\left(R_{1,2}^{k-1}\mathbf{x}_2^k\tilde{R}_{1,2}^{k-1}\right) + \left(R_{1,3}^{k-1}\mathbf{x}_3^k\tilde{R}_{1,3}^{k-1}\right)}{2} \quad (\text{B.1})$$

Finally, the observation vector describing the occluded marker m_1 in frame t , $\hat{\mathbf{x}}_1^k$, is calculated finding the nearest point on the circle to the point $\tilde{\mathbf{x}}_1^k$, as illustrated in figure B.1. The circle is the intersection of the two spheres, having as centre the marker positions \mathbf{x}_2^k , \mathbf{x}_3^k and as radii the distance between the

markers position \mathbf{x}_2^{k-1} , \mathbf{x}_3^{k-1} and the position of tracked marker \mathbf{x}_1^{k-1} in frame $k-1$, respectively.

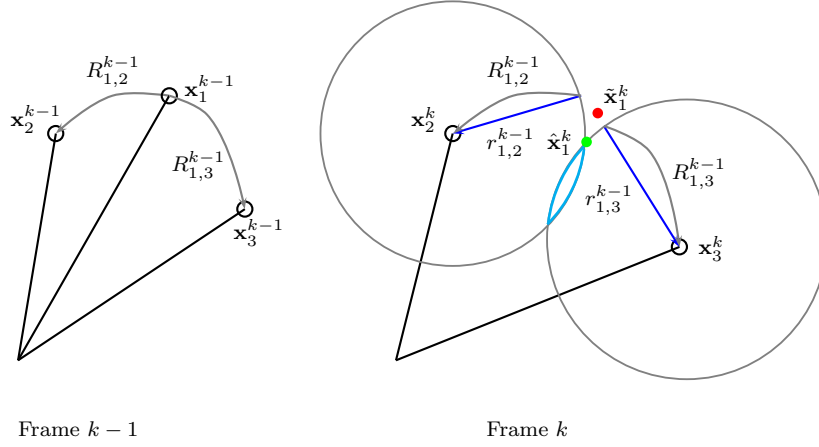


Figure B.1.: The calculation of the observation vector in the case where 2 markers are visible. Assuming that the rotation between each marker will vary slowly with time, we estimate the current position of the occluded marker. The red dot, $\tilde{\mathbf{x}}_1^k$, represents the mid-point between the estimated positions of the tracked marker, applying the rotors method to the visible markers \mathbf{x}_2^k and \mathbf{x}_3^k , respectively. The green dot, $\hat{\mathbf{x}}_1^k$, presents the shortest distance between the intersection of the 2 spheres and the red dot, and it is used as the observation vector in the Kalman filter $\mathbf{Z}_k = H\hat{\mathbf{x}}_1^k$.

The results are very similar to those in section 6.2. However, this method is not currently preferred because, in order to calculate the radii of the spheres r_{ij}^k , the distance D_{ij}^{k-1} ($D_{i,j}^{k-1} = |\mathbf{x}_j^{k-1} - \mathbf{x}_i^{k-1}|$) also needs to be calculated, thus adding more processing time.

C

Nearest point on circle from a point in space

This appendix considers the problem of finding a better approximation of the observation vector used in the Kalman filter (section 5.2), that is related to the position of the occluded marker. Using information from the rigid body, and also keeping in mind the fact that markers have a constant distance between each other, we can find the position of the non-visible marker. The observation vector describing the occluded marker m_1 in frame k , $\hat{\mathbf{x}}_1^k$, can be assigned as the shortest distance between the intersection of the two spheres with centre the positions of the markers m_2 and m_3 , radii the distances D_{12} and D_{13} , and the point in space, $\tilde{\mathbf{x}}_1^k$. The solution of this problem can be solved using conformal geometric algebra (CGA) as shown in section C.2. The intersection of 2 spheres is discussed in section 2.2.6. The distance from the point $\tilde{\mathbf{x}}_1^k$ (which can be considered as any point in space) to $\hat{\mathbf{x}}_1^k$ can be solved as a problem finding the nearest point from a circle to a point in space and it is utilised in section C.1.

C.1. The minimum distance from a circle and a point in space

The nearest point on a circle to a point in space can be also found using conformal geometric algebra. It is clear that the minimum distance is related to the projection of the point onto the plane Φ where the circle belongs. This can be

achieved by reflecting the point in the plane and finding the mid-point of the reflected and the original point. Hence, for circle $C = H(b) \wedge H(c) \wedge H(d)$ and point x with $X = H(x)$ (where the operation of $H(x)$ is the Hestenes' mapping in 2.2) we can formulate:

$$X' = \Phi X \Phi \quad (\text{C.1})$$

$$X^P = H\left(\frac{1}{2}(H^{-1}(X') + x)\right) \quad (\text{C.2})$$

A line, L , through this midpoint and the centre of the circle is then formed,

$$L = X^P \wedge C \wedge C \wedge n \quad (\text{C.3})$$

Intersecting this line with the circle will return two points which are the shortest and longest distances from the point in space. The selection of the appropriate point can be achieved using a simple distance comparison or using an oriented CGA (OCGA) formulation to pick the first intersection of the oriented line passing through the circle.

$$\begin{aligned} (X_1, X_2) &= L \vee C \\ X &= \arg(\max(X_1 \cdot X, X_2 \cdot X)) \end{aligned} \quad (\text{C.4})$$

This method is demonstrated in figure C.1.

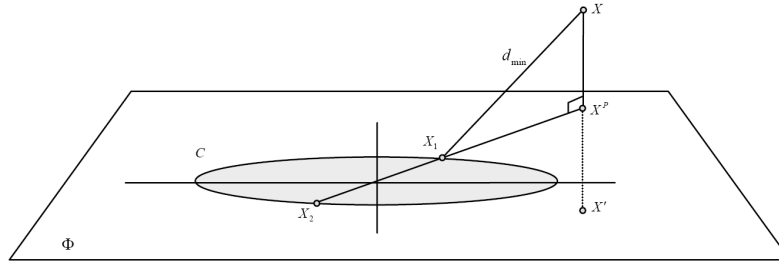


Figure C.1.: Finding nearest point on circle to point in space using CGA/OCGA.

C.2. The Solution

This section describes the solution of the problem. Assume that a sphere has centre \underline{c} and radius ρ . The spheres Σ_1 and Σ_2 can be expressed as blades in CGA using the formulation as reported in 2.2.5:

$$\Sigma = \left(c - \frac{1}{2}\rho^2 n\right) I$$

where

$$c = \frac{1}{2}c^2n + c - \frac{\bar{n}}{2} \quad (\text{C.5})$$

The intersection of the spheres is a circle or a single point, as in section 2.2.6.

$$C = \Sigma_1 \vee \Sigma_2 = [\langle \Sigma_1 \Sigma_2 \rangle_2] I \quad (\text{C.6})$$

The centre of the circle C can then be calculated using equation 2.77. Hence,

$$X = CnC$$

and the plane Φ that the circle belongs to can be formulated as:

$$\Phi = C \wedge n = \langle C n \rangle_4$$

Having the plane Φ and the point $y = \hat{\mathbf{x}}_1^t$ in the space, the nearest point on a circle, $\hat{\mathbf{x}}_1^t$, can be found from the methods in section C.1. Hence, $Y = H(y)$ and reflecting this in the plane Φ given Y' :

$$Y' = \Phi Y \Phi \quad (\text{C.7})$$

The mid-point Y^P is calculated as:

$$Y^P = H \left(\frac{1}{2} (H^{-1}(Y') + y) \right) \quad (\text{C.8})$$

Then, the line L is equal to:

$$L = Y^P \wedge CnC \wedge n = \langle \langle Y^P CnC \rangle_2 n \rangle_3 \quad (\text{C.9})$$

The intersection between line L and circle C will return us a bivector, AB , which represents the two possible values for $\hat{\mathbf{x}}_1^t$ (the minimum and maximum distances from the point in space).

$$AB = L \vee C \quad (\text{C.10})$$

Finally, the vectors A and B can be extracted from $A \wedge B$ using equations 2.65. Then, $\hat{\mathbf{x}}_1^t$ will be the value that returns the minimum distance from the point in space.

D

Nearest point on a line from a circle

This appendix considers the problem of finding the nearest point on a line from a circle in order to obtain a better estimate of missing marker position. In section 5.3.2 we presented an efficient model for estimating the missing marker on a limb segment, in case where 2 markers are visible on the same limb segment. Also, in section 5.3.5 we talked about instances where missing markers are visible in a single camera. This information identifies a line, L_1 , starting from the camera and passing through the position of the missing marker. Here we present an alternative way of estimating the missing marker which is visible to one camera, by finding the nearest point on line L_1 from circle C_1 . Circle C_1 is the circle created after the intersection of the spheres Σ_1 and Σ_2 , as described in section 5.3.2.

Consider a line L_1 passing through the points a and b , which are represented in CGA as:

$$A_1 = \frac{1}{2} (a^2 n + 2a - \bar{n}) \quad (\text{D.1})$$

$$B_1 = \frac{1}{2} (b^2 n + 2b - \bar{n}) \quad (\text{D.2})$$

Line L_1 will then be equal to:

$$L_1 = A_1 \wedge B_1 \wedge n = \langle \langle A_1 B_1 \rangle_2 n \rangle_3 \quad (\text{D.3})$$

Assume that circle C_1 lies on the plane Φ . Line L_1 intersect with plane Φ , otherwise L_1 and Φ are parallel. Thus,

$$B = \Phi \vee L_1 = [\langle \Phi L_1 \rangle_3]^* = \langle \Phi L_1 \rangle_3 I \quad (\text{D.4})$$

If $B^2 > 0$, then L_1 intersects with plane Φ , while if $B^2 = 0$, then L_1 and plane Φ are parallel. Firstly, consider the case where line L_1 and plane Φ intersect. In such a case the bivector B will be equal to $B = P \wedge n$ or $B = n \wedge P$. Vector P can be extracted from bivector B as discussed in section 2.2.5. Thereafter, we need to find the projection of the centre of the circle, $C = H(c)$ (where the operation of $H(x)$ is the Hestenes' mapping 2.2), on line L_1 . This is achieved by reflecting the centre of the circle over the line and then finding the mid point between point C and its reflection C' .

$$C' = L_1 C L_1 \quad (\text{D.5})$$

$$E = H \left(\frac{1}{2} (H^{-1}(C') + H^{-1}(C)) \right) \quad (\text{D.6})$$

We then need to find the point Y which is the projection of point E onto the plane Φ . A similar procedure as above will be formulated.

$$E' = \Phi E \Phi \quad (\text{D.7})$$

$$Y = H \left(\frac{1}{2} (H^{-1}(E') + H^{-1}(E)) \right) \quad (\text{D.8})$$

After finding the point Y , the algorithm can be separated into three possible subcases:

- If the distance from the centre of the circle, C , to that point, Y , is smaller than the radius of the circle, then a line $L_2 = P \wedge Y \wedge n$ will be established. The intersection of line L_2 with the circle C_1 will return 2 points, x_1 and x_2 , the shortest and longest distances from point P . The selection of the appropriate point can be achieved using a simple distance comparison.

$$(x_1, x_2) = L_2 \vee C = [\langle C L_2 \rangle_4]^* \quad (\text{D.9})$$

$$X = \arg(\max(x_1 \cdot P, x_2 \cdot P)) \quad (\text{D.10})$$

Assume that the point with the shortest distance from P is point x_1 . Obviously, the point on L_1 which is closer to x_1 is the projection of that point onto the line. Hence, the point we are looking for, x_n , is equal to:

$$x'_1 = L_1 x_1 L_1 \quad (\text{D.11})$$

$$x_n = H \left(\frac{1}{2} (H^{-1}(x'_1) + H^{-1}(x_1)) \right) \quad (\text{D.12})$$

Figures D.1 and D.2 illustrate the above. Point x_n is given in red.

However, there are some extreme instances to be considered:

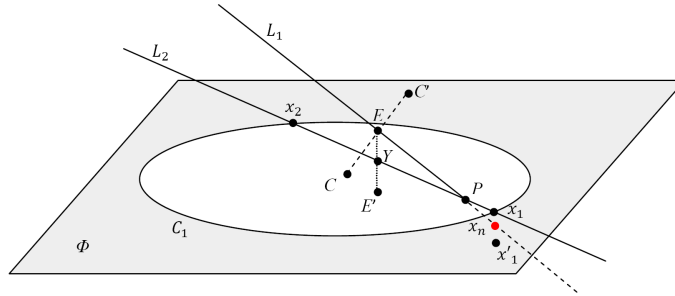


Figure D.1.: Finding the nearest point on a line from a circle using CGA. Case where the distance from the centre of the circle C_1 to that point Y is smaller than the radius of the circle and point P is inside the circle area.

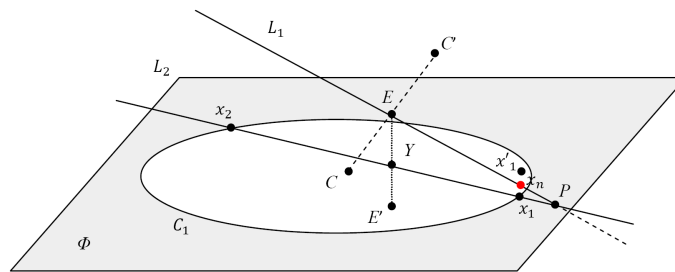


Figure D.2.: Finding the nearest point on a line from a circle using CGA. Case where the distance from the centre of the circle C_1 to that point Y is smaller than the radius of the circle and point P is outside the circle area.

- In the case where point P and Y are identical (Line L_1 is perpendicular to plane Φ), then $L_2 = C \wedge Y \wedge n$. The rest of the algorithm remains the same.
- In the case where line L_1 passes through the centre of the circle, line L_2 cannot be established using the above procedure. Hence, we consider line L_2 as the projection of line L_1 onto the plane Φ . The rest of the algorithm remains the same.
- In the case where line L_1 is perpendicular to plane Φ and also passes through the centre of the circle, then any point on circle C_1 has the same distance from line L_1 .
- If the distance from the centre of the circle, C , to that point, Y , is equal to the radius of the circle, then $x_1 = Y$. In such a case, the point on line we are looking for, x_n can be achieved by projecting the point x_1 onto the line L_1 as above.
- If the distance from the centre of the circle, C , to the point Y is greater

than the radius of the circle, then a line L_2 will be introduced which is equal to $L_2 = C \wedge Y \wedge n$. Hence, we have a similar situation as above. However, line L_2 is different. Again, the intersection of line L_2 with the circle C_1 will return 2 points, x_1 and x_2 , and the appropriate point can be selected as the point with the shortest distance from point P . Thus, point x_n is related to the projection of point x_1 onto line L_1 . Figure D.3 illustrates the above.

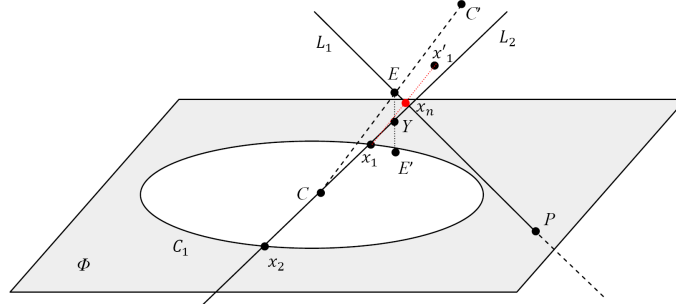


Figure D.3.: Finding the nearest point on a line from a circle using CGA. Case where the distance from the centre of the circle C_1 to that point Y is greater than the radius of the circle and point P .

In the second case, where $B^2 = 0$, plane Φ and line L_1 are parallel. We can check if the circle and the line are intersecting by calculating the *meet* between them.

$$G = C \vee L_1 = [\langle CL_1 \rangle_4]^* = \langle CL_1 \rangle_4 I \quad (\text{D.13})$$

Thus, there are three further possible subcases:

- If $G = 0$ then line L_1 belongs to plane Φ and intersects with the circle at two points. In such a case we can extract the two points as in section 2.2.5. Both points belong on line L_1 and on circle C .
- If $G = X$ and $X^2 = 0$ then line L_1 belongs to plane Φ and intersects with the circle at one point. That is the point we are looking for.
- If $G = X$ and $X^2 \neq 0$ then line L_1 and circle do not intersect at any point. This case can be further separated in two subcases; the instance where line L_1 belongs on the plane Φ and the instance where it does not.
 - In the case where line L_1 belongs to plane Φ , then the nearest point on the line from the circle can be achieved by finding the projection of the centre of the circle onto the line.
 - In the case where line L_1 does not belongs to plane Φ , then the problem can be solved by finding the projection of line L_1 onto the plane Φ . Hence, we can continue as above, having a line on plane Φ .

E

Nearest point on a line from a sphere

This appendix considers the problem of finding the nearest point on a line from a sphere. In section 5.3.3, an efficient model for estimating missing markers has been proposed for the case where 2 markers on a limb segment are occluded. However, there are instances where the missing marker is visible in a single camera, giving us more information about the position of this missing marker. This information identifies a line, L_1 , starting from the camera and passing through the position of the missing marker. Section 5.3.5 presents a reliable approach of estimating the missing markers for such instances. However, this chapter discusses an alternative way of estimating the occluded marker, which is visible to one camera, by finding the nearest point on line L_1 from sphere Σ_1 . Σ_1 has as its centre the position of the visible marker m_2 and radius the distance $|\hat{\mathbf{D}}_{j,2}^k|$, as calculated in section 5.3.3.

Consider a line L_1 passing through the points a and b , which are represented in CGA as:

$$A_1 = \frac{1}{2}(a^2n + 2a - \bar{n}) \quad (\text{E.1})$$

$$B_1 = \frac{1}{2}(b^2n + 2b - \bar{n}) \quad (\text{E.2})$$

Line L_1 is then:

$$L_1 = A_1 \wedge B_1 \wedge n = \langle \langle A_1 B_1 \rangle_2 n \rangle_3 \quad (\text{E.3})$$

We now need to find the projection of the centre of the sphere, C , onto line L_1 . This can be achieved by reflecting the centre of the circle in the line and then finding the mid point between the original, C , and the reflected point C' .

$$C' = L_1 C L_1 \quad (\text{E.4})$$

$$Y = H \left(\frac{1}{2} (H^{-1}(C') + H^{-1}(C)) \right) \quad (\text{E.5})$$

The problem can now be separated in three possible subcases; the case where line L_1 intersects the sphere at two points, the case where line L_1 touches the sphere and the case where the line L_1 and the sphere have no intersection points. The solution for all cases is given below:

- If the distance between the centre of the sphere and the point Y is greater than the radius of the sphere, then line L_1 does not intersect with sphere Σ_1 . Hence, Y represents the point on line we are looking for.
- If the distance between the centre of the sphere and the point Y is equal to the radius of the sphere, then line L_1 intersect with sphere Σ_1 at exactly one point, the point Y . Obviously, Y represents the point on line with minimum distance from the sphere.
- If the distance between the centre of the sphere and the point Y is smaller than the radius of the sphere, then line L_1 intersect with sphere Σ_1 at two points. Both points could represent the point we are looking for. However, we select as the position of the missing marker the point which returns minimum distance from the position in the previous frame.

List of Abbreviations

CGA	-	Conformal Geometric Algebra
CoR	-	Centre of Rotation
DOF	-	Degrees of Freedom
EKF	-	Extended Kalman Filter
GA	-	Geometric Algebra
GPLVN	-	Gaussian Process Latent Variable Model
KF	-	Kalman Filter
MoCap	-	Motion Capture
OCGA	-	Oriented Conformal Geometric Algebra

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] Chris Doran and Anthony Lasenby. *Geometric Algebra for Physicists*. Cambridge University Press, Cambridge UK, 2003.
- [16] Leo Dorst and Stephen Mann. Geometric algebra: A computational framework for geometrical applications (part 1). *IEEE Computer Graphics and Applications*, 22(3):24–31, 2002.
- [17] Stephen Mann and Leo Dorst. Geometric algebra: A computational framework for geometrical applications (part 2). *IEEE Computer Graphics and Applications*, 22(4):58–67, 2002.
- [18] Hermann Grassmann. *Ausdehnungslehre*. Enslin, 1862.
- [19] D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A unified language for mathematics and physics*. D. Reidel, 1984.
- [20] D. Hestenes. Old wine in new bottles: a new algebraic framework for computational geometry. In *Geometric Algebra with Applications in Science and Engineering*, pages 1–16. Birkhauser, 2001.
- [21] Andreas W. M. Dress and Timothy F. Havel. Distance geometry and geometric algebra. *Foundations of Physics*, 23(10):1357 – 1374, October 1993.
- [22] 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.
- [23] Jonathan Cameron. *Aspects of Geometric Algebra with applications in motion capture*. PhD thesis, Cambridge University Engineering Department, Cambridge, UK, 2007.
- [24] Jorge Stolfi. *Oriented projective geometry: a framework for geometric computations*. Academic Press, Boston, 1991.
- [25] PhaseSpace Inc.: Optical motion capture systems, <http://www.phasespace.com>.

- [26] CodaMotion. Optical motion capture systems, <http://www.codamotion.com>.
- [27] Vicon Motion System and Peak Performance. Motion capture systems, <http://www.vicon.com>.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] Berthold Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, April 1987.
- [34] 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.
- [35] 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.
- [36] Douglas J. Wiley and James K. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications*, 17(6):39–45, 1997.
- [37] Charles Rose, Michael Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
- [38] 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.
- [39] 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.

- [40] 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.
- [41] 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.
- [42] Klaus Dorfmueller-Ulhaas. Robust optical user motion tracking using a kalman filter. Technical Report TR-2003-6, Institut fuer Informatik, Universitaetsstr. 2, 86159 Augsburg, May 2003.
- [43] 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.
- [44] Lorna Herda, Pascal Fua, Ralf Plänkers, 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.
- [45] 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.
- [46] 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.
- [47] 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.
- [48] 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.
- [49] 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.
- [50] Guodong Liu and Leonard McMillan. Estimation of missing markers in human motion capture. *The Visual Computer*, 22(9-11):721–728, September 2006.
- [51] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME - Journal of Basic Engineering*, pages 35–45, 1960.

-
- [52] 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.
- [53] Andreas Aristidou, Jonathan Cameron, and Joan Lasenby. Predicting missing markers to drive real-time centre of rotation estimation. *Submitted to the 5th Conference on Articulated Motion and Deformable Objects, AMDO'08*, Mallorca, Spain, July 2008.