Articulated three-dimensional Human Modeling from Motion Capture Systems

JOÃO RENATO KAVAMOTO FAYAD

Dissertação para obtenção do Grau de Mestre em

ENGENHARIA BIOMÉDICA

Júri

Presidente:	Fernando Henrique Lopes da Silva, MD, PhD
Orientadores:	Pedro Manuel Quintas Aguiar, PhD
Vogais:	Alessio Del Bue, PhD
	Miguel Pedro Tavares da Silva, PhD

Outubro de 2008

Acknowledgements

It is with great pleasure that I take this opportunity to thank everyone that helped me, in a way or another, to conclude this work, as this thesis closes a five-year chapter of my life that would not have been possible without the help of many people.

First of all, I would like to thank Prof. Pedro Aguiar and Alessio Del Bue to have given me the opportunity to work with them on this subject. A special thank you goes to Alessio for all the support and motivation given almost on a daily basis through this last year. I am sure I grew both as student and a person, and all the eventual success of any future work of mine in this field will be due to this experience. I would also like to thank prof. Lopes da Silva for being available with such short notice to be my FML supervisor.

To all my classmates, for all the mutual support, companionship, daily lunches and pleasant times in and out of school, I wish them the best luck on their professional and personal lives. A special thanks will go to: Artur and Bruno ”Gilli”, my ”weapons of choice” for every group work, for the countless successful projects, reports and presentations the three of us did during this degree; Dárcio for all the amazing time we spent in Enschede, The Netherlands, as ERASMUS students; Everyone that went on our graduation trip to Cancun, as it certainly was one of the best weeks of my life.

To all of my friends, but specially to my flatmates, Artur, David and Lala (although they would make considerable amount of noise when I wanted to sleep) I owe a big thanks for all the good times we spent as everyone needs those moments to balance with the stressful student life. Again, even more special thanks to Artur for letting me sleep in his living room for this past month.

Last but not the least, I have to thank my family, without whom none of this would be possible. I thank my parents for always believing in my abilities and for all the effort they made so I could have the best education possible. I also have to thank all the time they spent doing my laundry, cooking my meals and driving me to Lisbon, as they were always making sure all I had to worry about were my studies. I just hope that at the end these five years they find it was worth it, and are proud my work.

Resumo

Modelos do Corpo Humano e a sua análise são, hoje em dia, utilizados em diversas áreas de aplicação, que vão da medicina à vigilância e segurança. Nesta tese, focamo-nos na criação automática de modelos biomecânicos para aplicações clínicas e no desporto.

Os mais modernos sistemas de captura de movimento são capazes de medir as coordenadas 3D de marcadores reflectores, colocados sobre a pele, com precisão considerada suficiente. Os actuais desafios que a análise da marcha enfrenta são a reprodutibilidade dos resultados, a criação de modelos personalizados, e a compensação dos artefactos causados pelos tecidos moles.

Desenvolvimentos recentes nas abordagens de *structure from motion* permitiram a extracção de estruturas rígidas articuladas com base em sequências de imagens 2D do seu movimento. Nesta tese, apresentamos um método automático para extrair os parâmetros das juntas mecânicas, que modelam as articulações humanas, a partir do das trajectórias 3D devolvidas pelos sistemas de captura de movimento. Descrevemos ainda uma nova abordagem que estima com maior precisão a componente rígida de um corpo não rígido, lidando ainda com a oclusão de alguns marcadores.

Para lidar com os artefactos causados pelos tecidos moles, é proposto um novo modelo quadrático para corpos deformáveis, definido como uma extensão dos modelos de corpo rígido existentes. Os parâmetros deste modelo são estimados com base em técnicas de optimização não-lineares.

Finalmente, a *performance* dos algoritmos é avaliada com base em dados sintéticos, e os resultados comparados aos valores reais dos parâmetros. Apresentamos também uma análise qualitativa dos algoritmos com base em dados reais.

Palávras-chave: Análise da Marcha, Modelos do Corpo Humanos, Factorização, Modelação não-linear, Biomecânica, Factorização.

Abstract

Human body models and their analysis are nowadays used in a wide range of applications, spanning from medicine to security and surveillance. In this thesis, we focus on the automatic creation of biomechanical human models for clinical and sports applications.

State of the art motion capture systems are able to measure the 3D coordinates of reflective markers placed above the skin with sufficient accuracy. The main issues clinical gait analysis is currently facing are the repeatability of the measurements, the creation of subject-specific models, and the compensation for soft-tissue movement.

Recent developments on structure from motion approaches have allowed the efficient recovery of articulated structures from a set of 2D images. In this thesis, we present a method to automatically recover joint parameters modelling the human articulations, from the 3D coordinates of a point cloud provided by motion capture systems. Additionally, we describe a novel approach capable of recovering a more accurate rigid body description of non-rigid bodies, and which is dealing as well with the problem of marker occlusions.

In order To deal with soft-tissue artifacts, we propose a new quadratic model for deformable bodies, as a natural extension of the existing rigid body models. The parameters of this model are estimated using a non-linear optimization technique.

Finally, we use synthetic data to assess the performance of the algorithms and compare the results with ground truth data. Qualitative analysis of real data sequences are also presented.

Keywords: Gait analysis, Human body models, Factorization, Non-linear modelling, Biomechanics.

Contents

List of Tables

List of Figures

Acronyms

AWGN	**-**	*Additive white Gaussian noise*
BA	**-**	*Bundle Adjustment*
MOCAP	**-**	*Motion Capture*
MRI	**-**	*Magnetic Resonance Imaging*
PCT	**-**	*Point Cloud Technique*
RMS	**-**	*Root Mean Square*
SfM	**-**	*Structure from Motion*
SVD	**-**	*Singular Value Decomposition*

Chapter 1

Introduction

A human body model is a mathematical description of its anthropometry, physiology or topology [1]. For instance, models can be built to test the human physiological response on crashworthiness tests [2], or to predict bone remodelling behaviour under different stress conditions [3]. In this thesis we focus on the computational modelling of the human body muskoloskeletal system *i.e.* on developing computational models of full body motion.

A Motion Capture (MOCAP) system is a device able to recover a full description of the motion on a scene, in order to analyse or transfer it to a digital model of the object performing the motion [4]. MOCAP systems are subdivided in mechanical, magnetic, ultrasonic or optical systems, being the categorization given by the physical principle by which the motion is detected and captured. Although the setup may be distinct among the different types of systems, they all have in common the fact that they detect the spatial location of a set of feature points in order to describe the motion of the scene. Given all the different types of MOCAP systems, optical systems have emerged as the primary choice for most of the applications as this setup provides a precision of the order of 1 mm for the 3D position of the feature points [5, 6, 4]. The most usual setup for optical MOCAP systems is composed of calibrated infrared cameras (at least 2, but typically more then 6) arranged around the area to be captured, and passive (reflective) markers, with diameter between 9 to 25 mm, placed on the surface of the object (see Figure 1.1) [6]. The output of the system is the set of 3D coordinates of the tracked markers over time.

The output of MOCAP systems is usually applied to animate previously existent digital models. However this approach not only requires great effort in building the model, but also limits the accuracy and versatility of the analysis, as it is bounded to adjust the existent model to the current application. In opposition, we will focus on the problem of *creating* a *human model* based on data acquired by MOCAP systems, as this approach guarantees subject specific modeling.

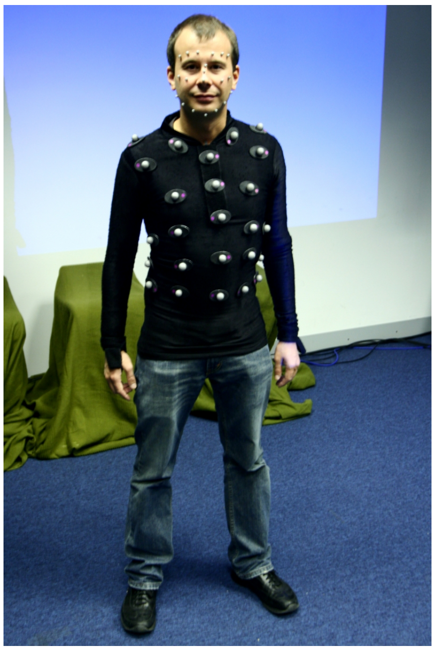


Figure 1.1: Example of a marker setup for a MOCAP analysis.

1.1 Motivation

Nowadays MOCAP systems are used to capture human motion in a variety of fields such as medicine [6], sports [7, 8], computer vision [9], character animation [10], or identification, security and surveillance [11]. In this thesis we will focus on biomechanical models of the human body *i.e.* on models that accurately describe the *motion* of the articulations, in order to predict the *forces* and *momenta* associated with it. This is in contrast with other applications such as character animation, where the focus is on the visual aspect of the model [12].

Biomechanics is the discipline that uses mechanical principles in order to study living organisms [7]. When applied to sports, it is a powerful tool that provides a qualitative and quantitative analysis in order to *improve performance* and to *prevent* or *treat injuries* [7]. In sports where the technique is a dominant factor, an analysis of the human motion can lead to an improvement of the performances, based on the information about articulations and muscles [13]. The information can also be used to evaluate the effect of a given motion of muscles and articulations, in order to prevent injuries or generate preventive and rehabilitative therapies [7].

In medicine, *gait analysis* is the main application of motion analysis tools. The study of the alterations in normal gait patterns are very important on areas such as cerebral palsy or prosthetic limbs, orthoses and total joint replacements, providing information both for diagnosis and treatment options [6, 14, 15].

When performing biomechanical studies on the human body, building accurate human models is one of the key steps for achieving meaningful results. When a generic study about a given motion is being conducted, the model can be built based on anthropometric data. However, when applying biomechanics in clinical cases or sports, models must be subject specific in order to have accurate results. In sports it is expected that only top athletes could have access to this technology in order to enhance their performances, thus their relatively small number can justify an extensive user intervention on the creation of individualised models. However, when we think about the universe of clinical patients that would benefit from this technology, the amount of resources dedicated just to building the custom models would be unbearable. Consequently, there is a need for finding methods to automatically create subject-specific reliable models.

One of the main problems of current methods for joint parameter estimation based on MOCAP systems is their *limited repeatability* [6]. A considerable amount of cases have been found where a subject is examined at two different laboratories, and the results differ significantly. This is a strong setback on the applications of these methods as accuracy and repeatability is of extreme importance in clinical analysis.

Another major source of error in MOCAP analysis are the *soft-tissue artifacts* [12, 16, 17]. In fact, although the relevant information about articulations used to build human models is given by the skeleton, the reflective markers used by MOCAP systems are placed above the skin. When examining a subject, there is an inherent relative motion between the soft-tissues surrounding the bone and the bone itself. These relative motions create artifacts on the data that degrade performances.

A key step in joint parameter estimation is the *model calibration i.e.* creating a model for the analysis that is subject-specific [6]. As mentioned above, this is essential to achieve accurate results. Some of the existing methods require accurate placement of reflective markers over some *specific* anatomical landmarks. Not only there is an inherent variability due to human error (different staff members will be placing the markers during the analysis), but also the landmarks are not easily found on all patients, depending on their medical conditions, *e.g., obesity* [6]. Thus, methods that do not rely on specific landmarks locations would be preferred.

In Summary, the main issues regarding gait analysis are the soft-tissue artifact, the subject specific modelling and the repeatability of the measurements. Our motivation is thus to tackle these issues, in order to have both a reliable tool for clinical applications, which can aid in diagnosis and rehabilitation, and a method that can be applied to sports, in order to enhance performances and prevent injuries. Nevertheless, we must not forget that MOCAP systems and human models are used in a wider range of applications that will also benefit from these improvements.

1.2 Prior work

As stated before, systems used to capture human motion are generally optical MOCAP systems, based on markers placed over the skin of the subject performing the motion. Most of the existent clinical systems for gait analysis use markers placed at specific anatomic landmarks. *Regression* techniques are then used to fit this data to a *Conventional Gait Model* (sometimes called *Helen Hayes* model), which models the hip, knee and ankle articulations as joints with 3 degrees of freedom [6]. As stated above, a disadvantage of this setup is the fact that it uses a small number of feature points that have to be accurately placed over anatomic landmarks, which in some cases are not very well defined in patients with certain medical conditions [6]. Besides, evidence has been brought up that most common equations used for regression provide unsatisfying results [18].

Other methods do not rely on regression techniques to compute the joint properties. For instance, some approaches use a method called *anatomical calibration* [6], which relies on a previously existent model of the joints. The joint parameters are then computed by fitting the data from the MOCAP systems, acquired while performing *predetermined* actions, to the existing model. For instance, markers attached to a body segment belonging to an articulation modelled as a ball and socket joint, would have their coordinates lying on a sphere centred at the joint centre. The joint centres are then computed by fitting the data to the ball and socket model using least-square optimization techniques [6] . Similar approaches can be used to model other articulations based on the choice of mechanical joint to use. The disadvantage of this method is that in some cases the performance of the test movements is impossible due to the medical conditions of the patients.

As mentioned in Section 1.1, one of the major sources of error in these measurements is the soft-tissue artifact, and so several attempts have been made to cope with this problem. Using optimization techniques to fit the motion capture data to a model, is one of them. For instance, attempts have been made to describe the real bone movement as a function of the observed soft-tissue movement, over all the range of motion [19, 20]. However, this approach suffers from the fact that the true bone motion is actually hard to define, as there is no consensus about a *gold-standard* for these measurements [6].

With the evolution of MOCAP systems, the number of feature points that can be tracked has increased since the first systems were created. This allowed the layout of new techniques such as point cluster techniques (PCTs). PCTs rely on the fact that, when using a higher number of features than the minimum number required, there is a redundancy in information, which will make soft-tissue artifacts less relevant [21]. This technique was later extended to a weighted algorithm, where points with higher deformations were given less weight when computing the joint parameters [19].

Recovering the structure of general objects based on information about their motion is a subject of interest in the field of computer vision, originating the family of algorithms designated *structure from motion* (SfM). These algorithms were first developed aiming to recover 3D shapes from a set of 2D images with multiple views of the scene. Although their motivation is not the same as ours, applications of such algorithms in medicine and sports can easily follow. From the different existing approaches we highlight the factorization-based approach as the one that has given more promising results.

Factorization methods for SfM are computational methods that use the unique rank properties of the measurement matrix to decompose it in a *motion* and *shape* factors. The first factorization methods for SfM were able to recover the shape and motion of a rigid object moving in a scene [22, 23]. Later, the approach was extended to recover structure and motion of various objects moving independently [24], and to deal with non-rigid objects with small deformations, using linear combinations of basis shapes [25, 26, 27, 28].

The next extension to factorization methods was made to deal with articulated objects. While the first approaches were based on fitting MOCAP data to previously existent models of the objects [29, 30], recent developments resulted on approaches that are able to infer articulated structure based solely on the motion data [31, 32]. Consequently, these approaches are of great interest when thinking about the creation of biomechanical models. To the best of our knowledge there are no applications of factorization methods for SfM in sports or medicine.

Although marker-based MOCAP systems are the most popular systems in clinical research, alternative approaches have also been considered. Among them, there are methods such as stereo radiography, bone pins, external fixation devices, and single or double plane fluoroscopy. However these methods are either invasive, or they limit motion range, or they require the exposure of the subject to radiation [6]. Models based on magnetic resonance imaging (MRI) have also been used [33, 34]. Although they can provide a detailed image of bones and muscles, this technique only works within small volumes and so their application is limited.

1.3 Objective

The objective of this work is to develop an algorithm to automatically compute biomechanical models of the human body based on the data provided by 3D MOCAP systems. We seek an algorithm that can be independent of the given MOCAP system's setup, only requiring a relatively high number of markers on each body segment. We also assume the segmentation of the data is known (*i.e.* to which body segment belongs a given set of points). The models should be able to deal with the main sources of error of the current systems: They should be subject specific; they should deal with soft-tissue artifacts; and they should be reasonably accurate in determining the joint parameters.

1.4 Proposed approach

Our method is a PCT where we assume the motion segmentation of each body limb to be known. Our approach is to recover the joint parameters using the recent developments on SfM algorithms for rigid articulated objects. Since we are normally dealing with non-rigid objects, we refine our algorithm by applying a variation of the weighted PCT, based on least-squares optimization, so that we have a first rigid approximation for each segment. Later, we use our own quadratic model for non-rigid bodies, initialized by the first estimate of the rigid segment, to compute a more accurate rigid component of the non-rigid segments. Finally, we combine the techniques for articulated SfM and our model for the non-rigid segments, to provide a final 3D articulated model of the human body.

We perform experiments using both synthetic and real data. Model validation is done solely by using synthetic data. Validating these kind of models with real data requires the knowledge of the real locations of joint centres and bone motions, which are not trivial to obtain since it requires specific equipment [12]. Therefore, validating with real data will be left as future work. However, we apply the algorithm to real data with the purpose of illustrating real applications of these algorithms, from which a first qualitative evaluation of performance can be done.

1.5 Original contributions

We emphasise the following original contributions:

- While the original SfM approaches are based on sequences of 2D images, in this work we present a consistent 3D re-formulation of the factorization approach for independent and articulated rigid bodies.
- We present a weighted factorization approach that is not only able to retrieve a more accurate rigid body description, but also it is able to deal with occlusion, which is one of the main issues on SfM algorithms.
- Additionally, we propose a new quadratic model to describe non-rigid bodies in order to model soft-tissues. With accurate modelling of soft-tissues we are able to retrieve a more accurate description of the rigid component of the movement (the bone) and provide more accurate articulation modelling.

Part of this work has been published in the Proceedings of the 8th International Symposium for Computer Methods on Biomechanics and Biomedical Engineering (CMBBE 2008) [35].

1.6 Thesis structure and organization

The thesis is structured as follows. Chapter 2 introduces the SfM factorization approach, with a detailed description of the independent and articulated rigid body factorization methods in the three dimensional case that we will apply for articulation modelling. We also present a weighted algorithm to retrieve a more accurate rigid body description when dealing with quasi-rigid objects and occlusion.

In Chapter 3 we propose a new quadratic model for non-rigid bodies. In order to compute the parameters for the quadratic model, we also present a Levenberg-Marquardt optimization scheme (generically termed as bundle-adjustment), that takes advantage of the particular characteristics of this model to achieve a more efficient computation.

Chapter 4 presents the MATLAB implementation of visualization and manual motion segmentation software tools. We briefly describe their functionalities and how they can help fulfilling the identification of each body part and their visualization in 3D.

In Chapter 5 we test all our algorithms providing a performance analysis for each of them. We also present real data applications of these algorithms for qualitative analysis and illustration purposes. Chapter 6 concludes this thesis with final considerations and directions for future work.

Chapter 2

Factorization Method for Structure from Motion

Factorization methods for structure from motion are a family of image based algorithms that model moving objects as a product of two factors: *motion* and *shape*. The shape parameters are defined as the 3D geometric properties of the object; the motion parameters are defined as the time-varying parameters of the motion (*e.g.* rotations and translations of the rigid body) that the shape performs in a metric space. One of the first factorization methods was proposed by Tomasi and Kanade [22]. This method successfully recovered camera motion and scene geometry (shape) based on a stream of 2D images. A certain number of feature points would be selected and tracked over the stream of images, providing the basis for the factorization algorithm. Our factorization approach, similarly to [22], assumes a set of P 3D feature points being tracked over F frames by a MOCAP system. This method relies on the key fact that 3D trajectories of points belonging to the same body share the same global properties.

The 3D trajectories provided by the MOCAP system can be arranged in a $3F \times P$ measurement matrix $\mathtt{W}$ as:

$$\mathtt{W} = \begin{bmatrix} \mathbf{w}_{11} & \dots & \mathbf{w}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{w}_{F1} & \dots & \mathbf{w}_{FP} \end{bmatrix}, \qquad (2.1)$$

where $\mathbf{w}_{ij}$ are the 3D coordinates of point j at frame i. Each 3D point $\mathbf{w}_{ij}$ can be written as:

$$\mathbf{w}_{ij} = \left[\begin{array}{c|c} \mathbf{r}_{1i}^T & t_{xi} \\ \mathbf{r}_{2i}^T & t_{yi} \\ \mathbf{r}_{3i}^T & t_{zi} \end{array}\right] \begin{pmatrix} x_j \\ y_j \\ z_j \\ 1 \end{pmatrix} = \left[\begin{array}{c|c} \mathtt{R}_i & \mathbf{t}_i \end{array}\right] \begin{bmatrix} \mathbf{s}_j \\ 1 \end{bmatrix}, \qquad (2.2)$$

where $\mathbf{s}_j$ is a 3-vector that has the $3D$ coordinates of point j, describing the shape, on a local referential; $\mathtt{R}_i$ and $\mathbf{t}_i$ are respectively the 3×3 rotation matrix and 3-vector of the translation parameters that

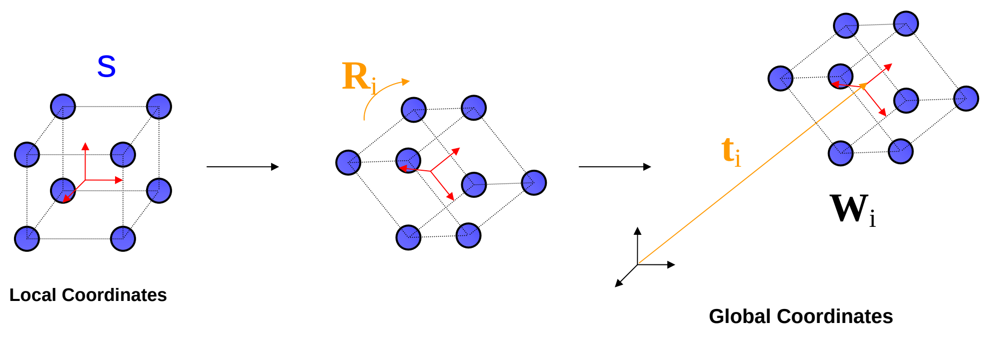


Figure 2.1: Graphical representation of the physical meaning of the motion and shape factors. The shape matrix $\mathtt{S}$ contains the 3D coordinates of the (blue) points that define the body, on the local (red) referential. The rotation matrix $\mathtt{R}_i$ and translation vector $\mathbf{t}_i$ represent the coordinate transformations that describe $\mathtt{S}$ on the global (black) referential, resulting in $\mathtt{W}_i$.

describe $\mathbf{s}_j$ on a global referential (see Figure 2.1). Stacking these equations for all the F frames and P points results in:

$$\mathtt{W} = \begin{bmatrix} \mathtt{W}_1 \\ \mathtt{W}_2 \\ \vdots \\ \mathtt{W}_F \end{bmatrix} = \begin{bmatrix} \mathtt{R}_1 \\ \mathtt{R}_2 \\ \vdots \\ \mathtt{R}_F \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_P \\ y_1 & y_2 & \cdots & y_P \\ z_1 & z_2 & \cdots & z_P \end{bmatrix} + \begin{bmatrix} \mathtt{T}_1 \\ \mathtt{T}_2 \\ \vdots \\ \mathtt{T}_F \end{bmatrix} = \mathtt{MS} + \mathtt{T}, \tag{2.3}$$

where $\mathtt{T}_i = \mathbf{t}_i \mathbf{1}_P^T$, with $\mathbf{1}_P^T$ being a P-vector with all entries equal to 1. The translational component $\mathbf{t}_i$ can be computed as the coordinates of the centroid of the point cloud at each frame $\mathtt{W}_i$. Thus, it can be easily eliminated by registering, at each frame, the point cloud to the origin *i.e* at each frame we subtract to the coordinates of every point the mean of the point cloud coordinates. In this scenario, it frequently occurs that, instead of $\mathtt{W}$, we consider a registered form of this matrix *i.e.* we use a matrix $\tilde{\mathtt{W}}$ such that:

$$\tilde{\mathtt{W}} = \mathtt{W} - \mathtt{T} = \mathtt{M}\,\mathtt{S}. \tag{2.4}$$

2.1 Rigid body

Let us consider the model defined in equation (2.3). Since $\mathtt{M}$ is a $3F \times 3$ matrix, and usually $F \gg 3$, by the properties of the $rank$ of a matrix, $rank(\mathtt{M}) \leq 3$. On the other hand, since $\mathtt{S}$ is a $3 \times P$ matrix, and usually $P \gg 3$, we also know that $rank(\mathtt{S}) \leq 3$. Although $\mathtt{T}$ is a $3F \times P$ matrix we know that all its columns are equal. Thus, as it only has one linearly independent column, $rank(\mathtt{T}) = 1$. From this considerations on the $rank$ of $\mathtt{M}$, $\mathtt{S}$ and $\mathtt{T}$ and by equation (2.3) we can now say that $rank(\mathtt{W}) \leq 4$. For the model defined by equation (2.4), since we have no translations, only $\mathtt{M}$ and $\mathtt{S}$ contribute to the $rank$ of $\tilde{\mathtt{W}}$ and so $rank(\tilde{\mathtt{W}}) \leq 3$. However the $rank$ properties are only valid in the ideal case with no noise.

When performing real experiments there will always be some noise which will increase the rank of $\tilde{\mathtt{W}}$. Noise can originate for instance from the MOCAP system's uncertainty in the position of the tracked feature points or some non-rigidity of the tracked objects.

Consider the singular value decomposition (SVD) of the registered matrix $\tilde{\mathtt{W}}$ defined by:

$$\tilde{\mathtt{W}} = \mathtt{U}_{3F\times 3F}\Sigma_{3F\times P}\mathtt{V}^T_{P\times P}, \tag{2.5}$$

where $\mathtt{U}$ and $\mathtt{V}$ are orthogonal matrices, and Σ is a diagonal matrix whose entries are the singular values σ_i of $\tilde{\mathtt{W}}$. Singular values are by definition non-negative ($\sigma_i \geq 0$) and are ordered in Σ, from top to bottom, in a decreasing way. Also there are as many positive singular values in a matrix as its rank *i.e.* if $r = rank(\tilde{\mathtt{W}}),\ \sigma_i > 0\ \forall i \leq r$.

The truncated SVD is a version of the decomposition that constraints the result to a $rank-k$ matrix. This is done by setting to zero all but the first k singular values. Consequently we can now use only the first k columns of $\mathtt{U}$ and $\mathtt{V}$ to compute the transformation. Mathematically, the $rank-k$ truncated SVD can be described as:

$$\hat{\mathtt{W}} = \mathtt{U}_k\,\Sigma_k\,\mathtt{V}_k^T, \tag{2.6}$$

where $\mathtt{U}_k$ is a $3F \times k$ matrix with only the first k columns of $\mathtt{U}$, $\mathtt{V}_k$ a $P \times k$ matrix with only the first k columns of $\mathtt{V}$, and Σ_k $k \times k$ the diagonal matrix with the first k diagonal entries of Σ. This result is the best approximation of $\tilde{\mathtt{W}}$ to a $rank-k$ matrix in the Frobenius norm sense. Since we know the ideal value for $rank(\tilde{\mathtt{W}})$ to be 3, we can use a $rank-3$ truncated SVD as a first global optimal fit to the measurements.

The $rank-3$ truncated SVD is not only useful in noise reduction but it can also be used the starting point for the factorization algorithm. Considering the expected dimensions of $\mathtt{M}$ and $\mathtt{S}$ we can compute a first estimation of these as:

$$\hat{\mathtt{M}} = \mathtt{U}_3\,\Sigma_3^{1/2}; \tag{2.7}$$

$$\hat{\mathtt{S}} = \Sigma_3^{1/2}\,\mathtt{V}_3^T. \tag{2.8}$$

However there exists an ambiguity in this factorization as any $\mathtt{A}$ 3×3 invertible matrix will satisfy the equality:

$$\hat{\mathtt{M}}\,\hat{\mathtt{S}} = \hat{\mathtt{M}}\,\mathtt{A}\,\mathtt{A}^{-1}\,\hat{\mathtt{S}}. \tag{2.9}$$

Being $\mathtt{A}$ an invertible matrix, it can be shown that it has an QR factorization *i.e* it can be factorized in the matrix product $\mathtt{A} = \mathtt{QR}$, where $\mathtt{R}$ is a 3×3 orthogonal matrix, and $\mathtt{Q}$ is a 3×3 upper triangular matrix. This implies that:

$$\mathtt{A}\,\mathtt{A}^{-1} = \mathtt{Q}\,\mathtt{R}\,\mathtt{R}^{-1}\,\mathtt{Q}^{-1} = \mathtt{Q}\,\mathtt{Q}^{-1}, \tag{2.10}$$

with the ambiguity being expressed in terms of the matrix product of an upper triangular matrix and its inverse. The initial factorization proposed in equations (2.7) and (2.8) do not guarantee that $\tilde{\mathtt{M}}$ is in fact a collection of F 3×3 rotation matrices. Thus the ambiguity stated in equation (2.10) is solved by finding the matrix $\mathtt{Q}$ that will transform each 3×3 matrix $\hat{\mathtt{M}}_i$ in a rotation (orthogonal) matrix $\mathtt{R}_i$. This can be achieved by imposing orthogonality constraints on $\hat{\mathtt{M}}_i\mathtt{Q}$, which is done by solving the set of linear equations for all the F frames:

$$\mathbf{m}_{ik}^T \, \mathtt{H} \, \mathbf{m}_{ik} = 1, \tag{2.11}$$

$$\mathbf{m}_{ik}^T \, \mathtt{H} \, \mathbf{m}_{il} = 0, \; l \neq k, \tag{2.12}$$

with $k, l = 1, 2, 3,$ $\mathbf{m}_{ik}$ and $\mathbf{m}_{il}$ are respectively the k-th and l-th row of matrix $\hat{\mathtt{M}}_i$, and $\mathtt{H} = \mathtt{Q}\,\mathtt{Q}^T$ is symmetric matrix (as $\mathtt{Q}$ is upper triangular). $\mathtt{Q}$ can thus be recovered from $\mathtt{H}$ by using Cholesky decomposition. We update the factorization in equations (2.7) and (2.8) to:

$$\mathtt{M} = \hat{\mathtt{M}}\,\mathtt{Q}; \tag{2.13}$$

$$\mathtt{S} = \mathtt{Q}^{-1}\,\hat{\mathtt{S}}. \tag{2.14}$$

While some algorithms solve this ambiguity in a frame-by-frame analysis, by using all the data available in $\tilde{\mathtt{W}}$ to compute $\mathtt{Q}$, we are actually taking in consideration all the frames to compute $\mathtt{S}$. In this way we can find the factors $\mathtt{M}$ and $\mathtt{S}$ that are more consistent with the whole motion. Finally, we can reconstruct $\hat{\mathtt{W}}$ as the product of the two parameters estimated by equations (2.13) and (2.14).

When the scene is composed of N rigid objects moving independently, the same considerations are valid. The model is simply expanded for each of the different independent objects, with $\mathtt{S}$ showing a block diagonal structure:

$$\tilde{\mathtt{W}} = \begin{bmatrix} \mathtt{M}_1 & \mathtt{M}_2 & \dots & \mathtt{M}_N \end{bmatrix} \begin{bmatrix} \mathtt{S}_1 & & & \\ & \mathtt{S}_2 & & \\ & & \ddots & \\ & & & \mathtt{S}_N \end{bmatrix}. \tag{2.15}$$

Thus, $rank(\tilde{\mathtt{W}}) \leq 3N$ when translations are not considered. If we consider translations, each model will increase their $rank$ by 1 dimension and so we will have $rank(\mathtt{W}) \leq 4N$ if translations are considered.

2.2 Articulated motion

As seen in section Section 2.1, when N rigid objects are moving independently, $rank(\tilde{\mathtt{W}}) \leq 3N$, in the case of the registered motion, and $rank(\mathtt{W}) \leq 4N$ when we consider translations. However if the rigid objects are linked by joints their motions are not independent and there is a loss in the degrees of freedom of the system. This constraint on the movement manifests itself in the measurement matrix

W as a decrease in rank [31, 36]. For the sake of simplicity we will only consider systems of two rigid bodies linked by a joint, despite the fact that these constraints are easily extended to a linked chain of rigid bodies.

2.2.1 Universal joint

By universal joint we man a kind of joint in which each of the two bodies is at a fixed distance to the joint centre, being the relative position of the bodies constrained, but their rotations remaining independent. In mechanics, this joint is usually denominated *spherical joint*. A scheme of this joint is presented in Figure 2.2.

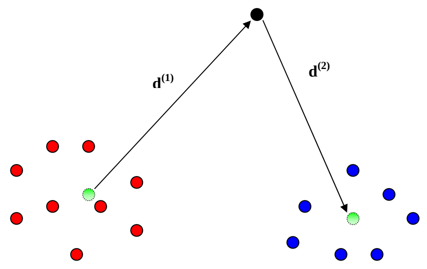


Figure 2.2: Scheme of a universal joint. The first body is represented by red points, while the second body is represented by blue points. The joint centre is shown as a black point. The 3-vector $\mathbf{d}^{(1)}$ stands for 3D coordinates of the joint centre in the local referential of the first body. The 3-vector $\mathbf{d}^{(2)}$ stands for 3D coordinates of the joint centre in the local referential of the second body.

Let $\mathbf{d}^{(1)} = [u,\, v,\, w]^T$ be the 3D coordinates of the joint centre in the local referential of the first body; $-\mathbf{d}^{(2)} = [u',\, v',\, w']^T$ be the 3D coordinates of the joint centre in the local referential of the second body; $\mathtt{R}^{(1)}$ and $\mathtt{R}^{(2)}$ the $3F \times 3$ matrices corresponding to a collection of 3×3 global rotation matrices over F frames, for the first and second body respectively; $\mathbf{t}^{(1)}$ and $\mathbf{t}^{(2)}$ the $3F$-vectors corresponding respectively to the first and second body global translation vectors.

The joint centre can thus be seen as a point that belongs to both bodies. In other words, its position can be described using the motion equations for the first and second body. With these considerations, a geometrical analysis of the joint structure reveals that:

$$\mathtt{R}^{(1)}\mathbf{d}^{(1)} + \mathbf{t}^{(1)} = -\mathtt{R}^{(2)}\mathbf{d}^{(2)} + \mathbf{t}^{(2)}. \tag{2.16}$$

Equation (2.16) is the mathematical formulation of the constraint of the universal joint. Thus we can write $\mathbf{t}^{(2)}$ as a function of $\mathbf{t}^{(1)}$ (or vice versa) which is equivalent to state that both 4D subspaces have a 1D intersection. The result of this consideration is that the measurement matrix of the universal joint must have $rank(\mathtt{W}) \leq 7$, one dimension less when comparing to the case of two independent rigid bodies. We are now able to factorize the measurement matrix as:

$$\mathtt{W} = \left[\begin{array}{c|c} \mathtt{W}^{(1)} & \mathtt{W}^{(2)} \end{array}\right] = \left[\begin{array}{ccc} \mathtt{R}^{(1)} & \mathtt{R}^{(2)} & \mathbf{t}^{(1)} \end{array}\right] \left[\begin{array}{cc} \mathtt{S}^{(1)} & \mathtt{D}^{(1)} \\ \mathtt{0}_{3\times P_1} & \mathtt{S}^{(2)} + \mathtt{D}^{(2)} \\ \mathbf{1}_{P_1}^T & \mathbf{1}_{P_2}^T \end{array}\right], \tag{2.17}$$

where $\mathtt{W}^{(1)}$ and $\mathtt{W}^{(2)}$ are respectively the measurement matrices for the first and second body; $\mathtt{D}^{(1)} = \mathbf{d}^{(1)}\,\mathbf{1}_{P_2}^T$ and $\mathtt{D}^{(2)} = \mathbf{d}^{(2)}\,\mathbf{1}_{P_2}^T$, where P_1 and P_2 are the number of points belonging the first and second body respectively, $\mathbf{1}_{P_1}$ a P_1-vector with all entries equal to 1 and $\mathbf{1}_{P_2}$ a P_2-vector with all entries equal to 1; $\mathtt{0}_{3\times P_1}$ is a $3 \times P_1$ zero matrix. Notice that in order to separate $\mathtt{W}^{(1)}$ from $\mathtt{W}^{(2)}$, we must assume the body segmentation to be known.

To recover the structure of the joint, one needs to find $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$. From equation (2.16) we can see that

$$\left[\mathtt{R}^{(1)}, \mathtt{R}^{(2)}, \mathbf{t}^{(2)} - \mathbf{t}^{(1)}\right] \left[\begin{array}{c} \mathbf{d}^{(1)} \\ \mathbf{d}^{(2)} \\ -1 \end{array}\right] = 0. \tag{2.18}$$

Therefore the joint parameters $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$ are easily computed once we have found the motion parameters $\mathtt{R}^{(1)}$, $\mathtt{R}^{(2)}$, $\mathbf{t}^{(1)}$ and $\mathbf{t}^{(2)}$. By using an approach similar to the one used in Section 2.1 for the independent rigid body, we first register each body to the origin of the global referential. As the universal joint constraint is described by a dependency between the translational components, the registered measurement matrix $\tilde{\mathtt{W}}$ is a $3F \times (P_1 + P_2)\ rank - 6$ matrix, and so:

$$\tilde{\mathtt{W}} = \left[\begin{array}{c|c} \mathtt{R}^{(1)} & \mathtt{R}^{(2)} \end{array}\right] \left[\begin{array}{cc} \mathtt{S}^{(1)} & 0 \\ 0 & \mathtt{S}^{(2)} \end{array}\right], \tag{2.19}$$

where $\mathtt{S}^{(1)}$ is a $3 \times P_1$ global shape matrix for the first body and $\mathtt{S}^{(2)}$ is a $3 \times P_2$ global shape matrix for the second body. The initial step in the factorization is then done by performing a truncated SVD with $k = 6$:

$$\tilde{\mathtt{W}} = \mathtt{U}_k\, \Sigma_k^{1/2}\, \Sigma_k^{1/2}\, \mathtt{V}_k^T = \left[\begin{array}{c|c} \mathtt{U}^{(1)} & \mathtt{U}^{(2)} \end{array}\right]_{3F\times 6} \left[\begin{array}{c|c} \mathtt{V}^{(1)} & \mathtt{V}^{(2)} \end{array}\right]_{6\times(P_1+P_2)}. \tag{2.20}$$

However the factorization is not final as $\left[\mathtt{V}^{(1)}|\mathtt{V}^{(2)}\right]$ is a dense matrix while the structure matrix defined in equation (2.19) has a specific structure. If we define an operator $N_l(.)$ that returns the left null-space of its argument, we can define a 6×6 transformation matrix $\mathtt{T}_U$ such that:

$$\mathtt{T}_U = \left[\begin{array}{c} N_l(\mathtt{V}^{(2)}) \\ N_l(\mathtt{V}^{(1)}) \end{array}\right]. \tag{2.21}$$

We can now recover $\mathtt{S}$ by pre-multiplying it by $\mathtt{T}_U$:

$$\mathtt{S} = \begin{bmatrix} N_l(\mathtt{V}^{(2)}) \\ N_l(\mathtt{V}^{(1)}) \end{bmatrix} \begin{bmatrix} \mathtt{V}^{(1)} \mid \mathtt{V}^{(2)} \end{bmatrix} = \begin{bmatrix} N_l(\mathtt{V}^{(2)})\,\mathtt{V}^{(1)} & N_l(\mathtt{V}^{(2)})\,\mathtt{V}^{(2)} \\ N_l(\mathtt{V}^{(1)})\,\mathtt{V}^{(1)} & N_l(\mathtt{V}^{(1)})\,\mathtt{V}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathtt{S}^{(1)} & 0 \\ 0 & \mathtt{S}^{(2)} \end{bmatrix}. \tag{2.22}$$

As we must keep the original data unaltered, we have to post-multiply $\left[\mathtt{U}^{(1)}|\mathtt{U}^{(2)}\right]$ by $\mathtt{T}_U^{-1}$:

$$\mathtt{M} = \begin{bmatrix} \mathtt{U}^{(1)} \mid \mathtt{U}^{(2)} \end{bmatrix} \begin{bmatrix} N_l(\mathtt{V}^{(2)}) \\ N_l(\mathtt{V}^{(1)}) \end{bmatrix}^{-1} = \begin{bmatrix} \mathtt{M}^{(1)} \mid \mathtt{M}^{(2)} \end{bmatrix}, \tag{2.23}$$

where $\mathtt{M}^{(1)}$ and $\mathtt{M}^{(2)}$ are respectively the motion matrix for the first and second body. Note that the ambiguity seen in equation (2.9) is still present here for each body, and so there is no guarantee that $\mathtt{M}^{(1)}$ or $\mathtt{M}^{(2)}$ are a collection of 3×3 rotation matrices. Due to the specific configuration of $\mathtt{S}$ seen in equation (2.19), there is no linear method to impose the orthogonality constraints to $\mathtt{M}$ while assuring that structure for $\mathtt{S}$. We chose to separate $\mathtt{W}^{(1)}$ and $\mathtt{W}^{(2)}$ treating them individually in the same way it was done for the independent rigid body in Section 2.1. Even though it is a suboptimal solution, as we are not using all the available data to solve the ambiguity, it is good approximation and it uses a simple linear form. Thus we apply in each case the transformation matrix $\mathtt{Q}$ as used before on equations (2.13) and (2.14).

After the estimation of the motion parameters we can finally solve the null-space problem stated in equation (2.18) to find the joint parameters $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$.

2.2.2 Hinge joint

In a hinge joint, two bodies can rotate around an axis such that the distance to that rotation axis is constant. Therefore their rotation matrices $\mathtt{R}^{(1)}$ and $\mathtt{R}^{(2)}$ are not completely independent. A scheme of the hinge joint is presented in Figure 2.3.

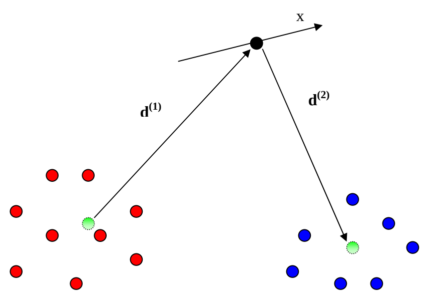


Figure 2.3: Scheme of a hinge joint. The first body is represented by red points, while the second body is represented by blue points. The joint centre is shown as a black point. The x-axis represents the rotation axis. The 3-vector $\mathbf{d}^{(1)}$ stands for the 3D coordinates of the joint centre in the local referential of the first body. The 3-vector $\mathbf{d}^{(2)}$ stands for the 3D coordinates of the joint centre in the local referential of the second body.

We keep the notation presented in Section 2.2.1 for the universal joint, where $\mathbf{d}^{(1)} = [u,\, v,\, w]^T$ are the 3D coordinates of the joint centre in the local referential of the first body, $-\mathbf{d}^{(2)} = [u',\, v',\, w']^T$ are the 3D coordinates of the joint centre in the local referential of the second body, $\mathtt{R}^{(1)}$ and $\mathtt{R}^{(2)}$ the $3F \times 3$ matrices corresponding to a collection of 3×3 global rotation matrices over F frames for the first and second body respectively, $\mathbf{t}^{(1)}$ and $\mathbf{t}^{(2)}$ the $3F$-vectors corresponding respectively to the first and second body global translation vectors.

By analysing the geometry of the joint, we can see that every vector belonging to any of the two bodies, that is parallel to the joint axis, must remain so throughout the movement. Let us choose an appropriate local referential, without loss of generality, where the axis of rotation of the joint is coincident with the x-axis. Let $\mathbf{e}_x = [1\,0\,0]^T$ be a vector representing the x-axis. Applying a general 3×3 rotation matrix $\mathtt{R} = [\mathbf{c}_1\,\mathbf{c}_2\,\mathbf{c}_3]$ to $\mathbf{e}_x$ will result in $\mathbf{c}_1 \cdot \mathbf{e}_x$ *i.e* the only column of the rotation matrix that affects vectors parallel to the x-axis is the first one. Therefore, to comply with the joint constraints, the first column of $\mathtt{R}^{(1)}$ must be equal to the first column of $\mathtt{R}^{(2)}$. We can now define the rotation matrices as $\mathtt{R}^{(1)} = [\mathbf{c}_1\,\mathbf{c}_2\,\mathbf{c}_3]$ and $\mathtt{R}^{(2)} = [\mathbf{c}_1\,\mathbf{c}_4\,\mathbf{c}_5]$. As all the points belonging to the rotation axis must fulfil both movement conditions, this results in a 2D intersection of the original 4D subspaces. Thus, when considering translations, $\mathtt{W}$ will then be given by:

$$\mathtt{W} = \left[\begin{array}{cccccc} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 & \mathbf{t}^{(1)} \end{array}\right] \left[\begin{array}{cccccc} x_1^{(1)} & \cdots & x_{P_1}^{(1)} & x_1^{(2)} & \cdots & x_{P_2}^{(2)} \\ y_1^{(1)} & \cdots & y_{P_1}^{(1)} & 0 & \cdots & 0 \\ z_1^{(1)} & \cdots & z_{P_1}^{(1)} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & y_1^{(2)} & \cdots & y_{P_2}^{(2)} \\ 0 & \cdots & 0 & z_1^{(2)} & \cdots & z_{P_2}^{(2)} \\ \multicolumn{3}{c}{\mathbf{1}_{P_2}^T} & \multicolumn{3}{c}{\mathbf{1}_{P_2}^T} \end{array}\right], \tag{2.24}$$

where, as defined in Section 2.2.1, P_1 and P_2 are the number of points belonging to the first and second body respectively, $\mathbf{1}_{P_1}$ a P_1-vector with all entries equal to 1 and $\mathbf{1}_{P_2}$ a P_2-vector with all entries equal to 1. In this case, $rank(\mathtt{W}) = 6$, if translations are not considered.

Since the constraints of this joint are limited only by the rotation matrices, it is possible to register the shapes to the origin of the referential without any influence on the constraints. By removing the translational factor in equation (2.24), we will get the $rank - 5$ system defined by:

$$\tilde{\mathtt{W}} = \left[\begin{array}{ccccc} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 \end{array}\right] \left[\begin{array}{cccccc} x_1^{(1)} & \cdots & x_{P_1}^{(1)} & x_1^{(2)} & \cdots & x_{P_2}^{(2)} \\ y_1^{(1)} & \cdots & y_{P_1}^{(1)} & 0 & \cdots & 0 \\ z_1^{(1)} & \cdots & z_{P_1}^{(1)} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & y_1^{(2)} & \cdots & y_{P_2}^{(2)} \\ 0 & \cdots & 0 & z_1^{(2)} & \cdots & z_{P_2}^{(2)} \end{array}\right]. \tag{2.25}$$

Once again we use the truncated SVD of $\tilde{\mathtt{W}}$ as the first step on the parameter estimation. For this case we use $k = 5$ in equation (2.6), giving a result similar to equation (2.20):

$$\tilde{\mathtt{W}} = \mathtt{U}_k\, \Sigma_k^{1/2}\, \Sigma_k^{1/2}\, \mathtt{V}_k^T = \left[\begin{array}{c|c} \mathtt{U}^{(1)} & \mathtt{U}^{(2)} \end{array}\right]_{3F \times 5} \left[\begin{array}{c|c} \mathtt{V}^{(1)} & \mathtt{V}^{(2)} \end{array}\right]_{5 \times (P_1 + P_2)}. \tag{2.26}$$

As we have seen in Section 2.2.1 this matrix $[\mathtt{V}^{(1)}|\mathtt{V}^{(2)}]$ is dense matrix, but what we need is to compute a matrix $\mathtt{S}$ with the structure defined in equation (2.25). Let $\mathtt{T}_H$ be a transformation matrix such that:

$$\mathtt{T}_H = \begin{bmatrix} \mathbf{b}^T \\ N_l(\mathtt{V}^{(2)}) \\ N_l(\mathtt{V}^{(1)}) \end{bmatrix}, \tag{2.27}$$

where $N_l(.)$ is the operator that returns the left null-space of its argument defined previously in Section 2.2.1, and $\mathbf{b}^T = [1\,0\,0\,0\,0\,0]$. By pre-multiplying $[\mathtt{V}^{(1)}|\mathtt{V}^{(2)}]$ with $\mathtt{T}_H$ we leave the first row intact and we zero-out some entries in order to get the structure presented in equation (2.25). Again, we need to post-multiply $[\mathbf{c}_1\,\mathbf{c}_2\,\mathbf{c}_3\,\mathbf{c}_4\,\mathbf{c}_5]$ with $\mathtt{T}_H^{-1}$ to keep the original data unaltered.

As observed in Section 2.1, in this approach arises an ambiguity. Following the same method as used in Section 2.2.1 for the universal joint, we separate $\mathtt{W}^{(1)}$ and $\mathtt{W}^{(2)}$ and use the transformation matrix $\mathtt{Q}$ as in equations (2.13) and (2.14) to solve for the ambiguity.

Now that we have the motion parameters, the last step is to recover the joint description. In the case of a hinge joint, the joint centre can lie anywhere on the axis of rotation. Still it must obey both motion equations *i.e.* equation (2.16) is still valid. Combining equation (2.16) with the properties of $\mathtt{R}^{(1)}$ and $\mathtt{R}^{(2)}$ for the hinge joint, the null-space problem defined by equation (2.18) can now be stated for this case as:

$$\left[\mathbf{c}_1\,,\mathbf{c}_2\,,\mathbf{c}_3\,,\mathbf{c}_4\,,\mathbf{c}_5\,,\mathbf{t}^{(2)} - \mathbf{t}^{(1)}\right] \begin{bmatrix} u+u' \\ v \\ w \\ v' \\ w' \\ -1 \end{bmatrix} = 0. \tag{2.28}$$

The null-space problem stated in equation (2.28) defines the coordinates of a point belonging to the rotation axis. Notice that we defined the axis of rotation aligned with the x-axis, so its direction is also known. Based on this knowledge we can now represent the rotation axis by a parametric equation of a line $l(\alpha)$ parallel to the x-axis that contains the joint centre:

$$l(\alpha) = [\mathbf{c}_1\,\mathbf{c}_2\,\mathbf{c}_3]\;[\alpha,\,v,\,w] + \mathbf{t}^{(1)}\;\forall\alpha \in \mathbb{R}. \tag{2.29}$$

2.3 Quasi-rigid objects and weighted factorization

The algorithms described in Section 2.1 and Section 2.2 solve the problem when the observed body is rigid. When dealing with non-rigid bodies they can still be used as a coarse rigid approximation of the data. As mentioned in Chapter 1, the meaningful information about how the human body articulates is given by modelling the skeleton, which, at this level of analysis, can be considered rigid. Still, MOCAP systems work based on capturing the 3D coordinates of markers placed above the skin, and are affected by the relative motions between soft-tissue and bone that happen while the subject is moving. Dealing with non-rigid bodies is thus one of the main challenges when developing algorithms for this purpose.

When using an SVD to estimate the motion and shape parameters, the resulting shape will be the one that minimises the error in a least-squares sense over all the frames. Nonetheless this might not be the best representation of the rigid component of the non-rigid shape. Factorizing with the previous algorithms can be seen as averaging the shape throughout the frames, resulting in an attenuation of the deformations. Inspired by previous approaches [37, 38, 39, 40], what we present here is an approach that uses a weighted SVD in order to penalise the contribution of the points which deform most. By doing so we will attenuate the contribution of the deformations, obtaining a more accurate rigid representation of the body.

2.3.1 The weighted factorization algorithm

When considering the weighted factorization approach, our goal is to find a better global rigid shape representation of a quasi-rigid body based on a penalisation of the points which deform the most. Let us assume for a moment that we know the best global rigid shape, and that its registered 3D coordinates over time are described by a matrix $\tilde{\mathtt{W}}^{(r)}$. Let the matrix $\tilde{\mathtt{W}}$ represent the data matrix resulting from tracking the quasi-rigid body with a MOCAP system, also registered. A measure of the non-rigidity of a given point on the matrix $\tilde{\mathtt{W}}$ can be given by how distant its trajectory is from the best rigid description given by $\tilde{\mathtt{W}}^{(r)}$. Note that the number of point trajectories described by each matrix is the same and there is a direct correspondence between them, as they refer to the same body. Based on this idea, we will rearrange the data matrix defined in equation (2.1) as:

$$\tilde{\mathtt{W}} = \begin{bmatrix} \tilde{\mathbf{w}}_{11}^T & \tilde{\mathbf{w}}_{12}^T & \dots & \tilde{\mathbf{w}}_{1P}^T \\ \tilde{\mathbf{w}}_{21}^T & \tilde{\mathbf{w}}_{22}^T & \dots & \tilde{\mathbf{w}}_{2P}^T \\ \vdots & \vdots & & \vdots \\ \tilde{\mathbf{w}}_{F1}^T & \tilde{\mathbf{w}}_{F2}^T & \dots & \tilde{\mathbf{w}}_{FP}^T \end{bmatrix}, \tag{2.30}$$

where $\tilde{\mathtt{W}}$ is an $F \times 3P$ matrix, with P is the number of feature points tracked over F frames by the MOCAP system. The matrix corresponding to the best global rigid shape $\tilde{\mathtt{W}}^{(r)}$ will be arranged similarly as in equation (2.30). The 3D trajectories of a generic point j are thus described in the $F \times 3$ matrix $\tilde{\mathtt{W}}_j$ defined by:

$$\tilde{\mathtt{W}}_j = \begin{bmatrix} \tilde{\mathbf{w}}_{1j}^T \\ \tilde{\mathbf{w}}_{2j}^T \\ \vdots \\ \tilde{\mathbf{w}}_{Fj}^T \end{bmatrix}, \tag{2.31}$$

with $j = 1, \ldots, P$. We define $\tilde{\mathtt{W}}_j^{(r)}$ as the 3D trajectories of the same generic point j in the best global rigid shape description. We can now define an error matrix $\mathtt{E}_j$ as:

$$\mathtt{E}_j = \tilde{\mathtt{W}}_j^{(r)} - \tilde{\mathtt{W}}_j, \tag{2.32}$$

with $\mathtt{E}_j$ an $F \times 3$ matrix and $j = 1, \ldots, P$. As described above, this matrix indicates how distant the rigid and non-rigid trajectories of a generic point j are. Thus, for deformable points, $||\mathtt{E}_j||$ will be higher than for rigid points. A weight matrix that assigns higher weights to rigid points and lower weights to deformable points can now be defined as:

$$\mathtt{C}_j = \mathrm{cov}(\mathtt{E}_j)^{-1}, \tag{2.33}$$

as deformable points are bound to originate higher covariance values. Given this weight matrix, a better rigid description of the deformable body can be found by solving the least-squares problem given by:

$$\underset{\mathtt{M}_i, \mathbf{s}_j}{\arg\min} \sum_{i=1}^{F} \sum_{j=1}^{P} (\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{M}_i \mathbf{s}_j)^T \, \mathtt{C}_j \, (\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{M}_i \mathbf{s}_j). \tag{2.34}$$

What we propose here is a two-step iterative algorithm that, from an initial estimation of the data for the best rigid shape, will compute a better global rigid shape description based on equation (2.34). If we have an estimation for $\tilde{\mathtt{W}}^{(r)}$ the weight matrix $\mathtt{C}_j$ can be computed. However, neither $\tilde{\mathtt{W}}^{(r)}$ nor $\mathtt{C}_j$ are known. Thus, the registered measurement matrix $\tilde{\mathtt{W}}$ will be used as an estimation of $\tilde{\mathtt{W}}^{(r)}$. With this, we intend to find the factors $\mathtt{M}$ and $\mathtt{S}$ that minimise the Frobenius distance of the measurement matrix to the weighted rigid body description. We will now rewrite equation (2.34) as:

$$\underset{\mathtt{M}_i, \mathbf{s}_j}{\arg\min} \sum_{i=1}^{F} \sum_{j=1}^{P} (\tilde{\mathbf{w}}_{ij} - \mathtt{M}_i \mathbf{s}_j)^T \, \mathtt{C}_j \, (\tilde{\mathbf{w}}_{ij} - \mathtt{M}_i \mathbf{s}_j). \tag{2.35}$$

Still equation (2.35) is not trivial to solve as it is a minimisation of two parameters. Let us assume we also know an estimation of $\mathtt{M}$. We can now find a solution for $\mathtt{S}$ by rearranging equation (2.35) as (for more details see Appendix A):

$$\mathbf{s}_j = (\sum_{i=1}^{F} \mathtt{M}_i^T \, \mathtt{C}_j \, \mathtt{M}_i)^{-1} \sum_{i=1}^{F} \mathtt{M}_i \tilde{\mathbf{w}}_{ij}. \tag{2.36}$$

This equation computes $\mathtt{S}$ based only in matrix products and matrix inversions and so it is done with ease.

On the other hand, if we assume $\mathtt{S}$ to be known , a similar solution can be found for $\mathtt{M}$. Nonetheless, we must first rearrange $\mathtt{M}_i$ into a 9-vector $\mathbf{m}_i$ defined by:

$$\mathbf{m}_i = \begin{bmatrix} \mathbf{r}_{1i} \\ \mathbf{r}_{2i} \\ \mathbf{r}_{3i} \end{bmatrix}, \tag{2.37}$$

and also rearrange $\mathbf{s}_j$ into a 3×9 block diagonal matrix $\mathtt{S}_j$ defined by:

$$\mathtt{S}_j = \begin{bmatrix} \mathbf{s}_j^T & & \\ & \mathbf{s}_j^T & \\ & & \mathbf{s}_j^T \end{bmatrix}. \tag{2.38}$$

Based on equation (2.35) we can now compute each vector $\mathbf{m}_i$ as (for more details see Appendix A):

$$\mathbf{m}_i = (\sum_{j=1}^{P} \mathtt{S}_j^T \, \mathtt{C}_j \, \mathtt{S}_j)^{-1} \sum_{j=1}^{P} \mathtt{S}_j \, \tilde{\mathbf{w}}_{ij}. \tag{2.39}$$

Each 9-vector $\mathbf{m}_i$ can now be rearranged into a 3×3 matrix $\mathtt{M}_i$. However there is again no guarantee that $\mathtt{M}_i$ will be a rotation matrix. We chose to project each known affine matrix $\mathtt{M}_i$ into its closest rotation matrix. This can be done optimally by decomposing each matrix $\mathtt{M}_i$ using an SVD ($\mathtt{M}_i \overset{SVD}{=} \mathtt{U} \, \Sigma \, \mathtt{V}^T$) and imposing $\Sigma = \mathtt{I}_{3\times3}$, where $\mathtt{I}_{3\times3}$ is the identity matrix [41]. If we denote the projection by $\hat{\mathtt{M}}_i$, it can be defined as:

$$\hat{\mathtt{M}}_i = \mathtt{U} \, \mathtt{V}^T. \tag{2.40}$$

Equations (2.36) and (2.39) naturally form an iterative method for the computation of M and S as the inputs of one are the outputs of the other. All we need now is an initial estimate of W to compute the weight matrix $\mathtt{C}_j$, an initial estimate of M and S, and a stoppage criterion.

For the initial estimations of W, M and S we will use the aforementioned rigid body factorization defined in Section 2.1. As this factorization gives us an approximation of a rigid body motion, it can be a good initialization for this algorithm. For the stopping criterion, we have chosen to use the convergence of the Frobenius norm of the global error matrix E defined by:

$$||\mathtt{E}|| = ||[\mathtt{E}_1 \, \mathtt{E}_2 \, \cdots \, \mathtt{E}_P]||\,. \tag{2.41}$$

Finally the algorithm can be summarised into the following steps:

1. Initialization: Compute the $rank-3$ approximation of W and factorize into M and S using the method described in Section 2.1.

2. With the current estimations of M and S compute the weight matrices $\mathtt{C}_j$ using equation (2.33).

3. Using the current estimation of M, compute S by using equation (2.36).

4. Based on the current estimation of S from Step 2, compute M by using equation (2.39).

5. Apply the orthogonality constrains to $\mathtt{M}_i$ defined in equation (2.40).

6. Repeat Steps 2 to 4 until convergence of the Frobenius norm of E is reached.

2.3.2 Weighted factorization and translation

In Section 2.3.1 we defined a weighted algorithm to estimate a better rigid representation of the global shape, based on the registered data. Still, if the deformation is strongly directional (*e.g.* muscular contraction) the rigid translation may be biased towards the deformation direction. Here we present a new version of the weighted factorization algorithm summarised at the end of Section 2.3.1, that incorporates an estimation for the translation. Since translation is not part of the global shape parameters, we must start to modify our previous algorithm in the computation of $\mathtt{M}$. Based on equation (2.2) we can update equation (2.37) to:

$$\mathbf{m}_i = \begin{bmatrix} \mathbf{r}_{1i} \\ \mathbf{t}_{xi} \\ \mathbf{r}_{2i} \\ \mathbf{t}_{yi} \\ \mathbf{r}_{3i} \\ \mathbf{t}_{zi} \end{bmatrix}, \tag{2.42}$$

where $\mathbf{m}_i$ is now a 12-vector; and update equation (2.38) to:

$$\mathtt{S}_j = \begin{bmatrix} \mathbf{s}_j^T\,1 & & \\ & \mathbf{s}_j^T\,1 & \\ & & \mathbf{s}_j^T\,1 \end{bmatrix}, \tag{2.43}$$

where $\mathtt{S}_j$ is now a 3×12 block diagonal matrix. Now we can update equation (2.39) to use the unregistered data matrix $\mathtt{W}$:

$$\mathbf{m}_i = (\sum_{j=1}^{P} \mathtt{S}_j^T\,\mathtt{C}_j\,\mathtt{S}_j)^{-1} \sum_{j=1}^{P} \mathtt{S}_j\,\mathbf{w}_{ij}, \tag{2.44}$$

where $\mathtt{S}_j$ is defined by equation (2.43) and $\mathbf{m}_i$ defined by equation (2.42).

The estimation of the global shape parameters is still done by equation (2.36) using the registered data matrix $\tilde{\mathtt{W}}$. However, since we have defined a new way to compute the translation, this registration is made by subtracting to every 3D point coordinates at each frame, not the mean of the point cloud, but the new translational component $\mathbf{t}_i = [\ t_{xi} \quad t_{yi} \quad t_{zi}\ ]^T$ computed in equation (2.44).

Summarising, the weighted algorithm to compute a better representation for the global shape parameters and estimate the rotations and translations of the motion can be described by the following steps:

1. Initialization: Compute the initial estimations for $\mathtt{M}$, $\mathtt{S}$ and $\mathbf{t}$ using the rigid body factorization method described in 2.1.

2. Use the current estimation of $\mathbf{t}$ to register the data matrix $\mathtt{W}$.

3. With the current estimation of $\mathtt{M}$ and the registered matrix $\tilde{\mathtt{W}}$ computed in Step 2, compute a new estimation for $\mathtt{S}$ using equation (2.36)

4. Based on the estimation of $\mathtt{S}$ computed in Step 3, compute a new estimation of $\mathtt{M}$ and $\mathbf{t}$ based on equation (2.44).

5. Repeat Steps 2 to 5 until convergence of the Frobenius norm of ε is achieved.

2.3.3 Weighted factorization with occlusion

When using MOCAP systems one of the problems that might occur is the *occlusion* of the feature points *i.e.* in some of the frames there might not be any data available for some markers (and that is why this problem is also named *missing data*). This occlusion occurs due to problems with the markers (*e.g.* the marker detaching from the body being tracked), problems with the tracking system itself (*e.g.* the body moving out of the range of the system) or when the marker is covered from the cameras by the body in motion. When occlusion happens the measurement matrix will contain, in some frames, fewer 3D point coordinates. Thus the global properties of the system will be altered, causing the algorithm to collapse. What we present here is an update of the algorithm that we began discussing in Section 2.3.1 and Section 2.3.2 to handle cases of occlusion. However we assume that we know exactly which points are missing when it occurs.

Even though some points may be occluded in a given frame, equation (2.2) still holds true for the points that are not occluded, as do all the rank considerations made in Section 2.1. Using these considerations, the approach presented in the previous sections can be easily modified. If a given feature point j was occluded at frame i, then the 3D coordinates $\mathbf{w}_{ij}$ will be missing. Since we do not have any information about missing markers we simply ignore its contribution to the computations of $\mathtt{M}$, $\mathtt{S}$ and $\mathbf{t}$, and use only the available information. Let us define a $F \times P$ binary matrix $\mathtt{Z}$ such that:

$$z_{ij} = \begin{cases} 1, & \text{if } \mathbf{w}_{\mathrm{ij}} \, \mathrm{is\,available.} \\ 0, & \text{if } \mathbf{w}_{\mathrm{ij}} \, \mathrm{is\,occluded.} \end{cases} \tag{2.45}$$

Now all we need to do is use $\mathtt{Z}$ to set to zero the contributions of the missing data. This can be done by updating equations (2.44) and (2.36) as:

$$\mathbf{m}_i = (\sum_{j=1}^{P} z_{ij}\, \mathtt{S}_j^T\, \mathtt{C}_j\, \mathtt{S}_j)^{-1} \sum_{j=1}^{P} z_{ij}\, \mathtt{S}_j\, \mathbf{w}_{ij}; \tag{2.46}$$

$$\mathbf{s}_j = (\sum_{i=1}^{F} z_{ij}\, \mathtt{M}_i^T\, \mathtt{C}_j\, \mathtt{M}_i)^{-1} \sum_{i=1}^{F} z_{ij}\, \mathtt{M}_i \tilde{\mathbf{w}}_{ij}. \tag{2.47}$$

When $\mathbf{w}_{ij}$ is missing, $\mathtt{E}_j$ can still be computed. However, entries corresponding to missing data will not be used on the computations, making $\mathtt{C}_j$ independent of missing data.

From equation (2.46) we can see that if $\mathtt{S}$ is known then, even if occlusion occurs in some frames,

$\mathbf{m}_i$ can still be estimated. In a similar way $\mathbf{s}_j$ is computed based on all the information available in $\mathtt{W}_i$, and thus it can also be estimated. Clearly, there is a limit on the amount of data that can be occluded for the algorithm to work. However finding a theoretical limit is not trivial, being this usually done with experimental results. The resulting algorithm has the same outline as the one described at the end of Section 2.3.2, but with equations (2.44) and (2.36) being replaced respectively by (2.46) and (2.47).

Chapter 3

Quadratic Model for Deformable Bodies

The factorization methods studied on Chapter 2 are based on the assumption that we are dealing with rigid bodies. When applied to deformable bodies, these algorithms make approximations to find the best rigid body description. As mentioned in Chapter 1, one of the main issues regarding accurate joint parameter estimation in humans is the soft-tissue artifact. At this level of analysis, bones can be seen as rigid bodies when compared to soft-tissues. Thus, if we are able to exactly model non-rigid bodies, we will be able to correctly separate the rigid contributions (skeleton) from the deforming one (soft-tissue). Consequently, a more accurate estimation of the motion of the bones composing the articulations will be possible, leading to a more accurate joint parameter estimation.

One of the most popular approaches when modelling deformable bodies is to approximate them as a linear combination of different rigid basis shapes [42, 25]. However, deformations occurring on the human body due to soft-tissue tend to have quadratic behaviour (*e.g.* muscle contractions), increasing considerably the number of basis shapes required to accurately approximate the deformations. This has a negative impact on computational costs, allowing us to use only a limited number of basis shapes, and thus compromising accuracy. Moreover, due to the high number of parameters to estimate, it is common to obtain various local minima when applying minimisation schemes to solve this problem, thus decreasing accuracy.

If we are to deal with quadratic deformations, the most logical approach leads to use a quadratic model for deformable bodies. Inspired by previous works on the field of computer graphics [10, 43], we present a new quadratic model for non-rigid bodies using geometric constraints, built as an extension to the factorization-based rigid body model as described in Section 2.1.

3.1 Quadratic model formulation

Our model for deformable bodies expands the rigid body formulation defined by equation (2.3), to a formulation that uses linear, quadratic and crossed-terms of the previous rigid shape matrix. Let us define the new shape matrix as:

$$\mathtt{S} = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_P \\ y_1 & y_2 & \cdots & y_P \\ z_1 & z_2 & \cdots & z_P \\ \hline x_1^2 & x_2^2 & \cdots & x_P^2 \\ y_1^2 & y_2^2 & \cdots & y_P^2 \\ z_1^2 & z_2^2 & \cdots & z_P^2 \\ \hline x_1y_1 & x_2y_2 & \cdots & x_Py_P \\ y_1z_1 & y_2z_2 & \cdots & y_Pz_P \\ z_1x_1 & z_2x_2 & \cdots & z_Px_P \end{array}\right] = \left[\begin{array}{c} \mathtt{S}^{(\Gamma)} \\ \hline \mathtt{S}^{(\Omega)} \\ \hline \mathtt{S}^{(\Lambda)} \end{array}\right], \tag{3.1}$$

where $\mathtt{S}^{(\Gamma)}$ is the $3 \times P$ linear shape matrix, $\mathtt{S}^{(\Omega)}$ the $3 \times P$ quadratic shape matrix and $\mathtt{S}^{(\Lambda)}$ is the $3 \times P$ cross-values shape matrix. Given this new structure of $\mathtt{S}$, we introduce the motion matrix $\mathtt{M}_i$ defined by:

$$\mathtt{M}_i = \mathtt{R}_i \left[\begin{array}{ccc} \Gamma_i & \Omega_i & \Lambda_i \end{array}\right], \tag{3.2}$$

where $\mathtt{R}_i$ is a 3×3 rotation matrix, and Γ_i is a 3×3 transformation matrix associated with linear deformations, Ω_i is a 3×3 transformation matrix associated with quadratic transformations and Λ_i is a 3×3 transformation associated with cross-values deformations. By modeling the rotations in a separate matrix $\mathtt{R}_i$, we are defining the deformation matrices in the local referential of the body. Using the same formulation as in the model for rigid bodies, and stacking the equation for all the F frames, we can now define:

$$\tilde{\mathtt{W}} = \left[\begin{array}{cccc} \mathtt{R}_1 & & & \\ & \mathtt{R}_2 & & \\ & & \ddots & \\ & & & \mathtt{R}_F \end{array}\right] \left[\begin{array}{ccc} \Gamma_1 & \Omega_1 & \Lambda_1 \\ \Gamma_2 & \Omega_2 & \Lambda_2 \\ \vdots & \vdots & \vdots \\ \Gamma_F & \Omega_F & \Lambda_F \end{array}\right] \left[\begin{array}{c} \mathtt{S}^{(\Gamma)} \\ \mathtt{S}^{(\Omega)} \\ \mathtt{S}^{(\Lambda)} \end{array}\right] = \mathtt{M}\,\mathtt{S}, \tag{3.3}$$

where $\tilde{\mathtt{W}}_i$ is the data matrix containing the 3D coordinates of the feature points, registered to the origin of the global referential. Following similar considerations as done in Chapter 2, this rigid model is described by a $rank(\tilde{\mathtt{W}}) \leq 9$ constraint.

This quadratic model is in fact an extension of the linear rigid body model defined in Section 2.1 to deal with quadratic and cross-value terms, while keeping the same factorization into *motion* and *shape* factors. Note that a rigid body is still easily expressed by this model if we make $\Gamma_i = \mathtt{I}_{3\times3}$, $\Omega_i = \mathtt{0}_{3\times3}$ and $\Lambda_i = \mathtt{0}_{3\times3}$ for every frame i, where $\mathtt{I}_{3\times3}$ is the 3×3 identity matrix, and $\mathtt{0}_{3\times3}$ is a 3×3 zero matrix. Accordingly, the rank constraints in this case will be still satisfied, giving $rank(\tilde{\mathtt{W}}) \leq 3$.

By combining the new shape matrices with the associated transformation matrices, we are able to model characteristic soft-tissue motions such as bending, bulging, jiggling or stretching. Detailed description about the role of the different matrices will be addressed in the following sections. For the sake of notation simplicity, we will only consider one frame of the motion, and the i index will be dropped. However, results can be easily extended to a general case.

3.1.1 Linear deformation and shape matrices

Before we can analyse the role of the linear deformation matrix Γ, a few considerations about the model must be done. Every full-rank 3×3 matrix can be expressed with a RQ decomposition, from which results a rotation matrix $\mathtt{R}$ and an upper triangular matrix Ω. Since in our model rotations are fully given by $\mathtt{R}_i$, a RQ decomposition of $[\Gamma\,\Omega\,\Lambda]$ should not incorporate a rotation component *i.e* the rotation matrix of that decomposition should be the 3×3 identify matrix $\mathtt{I}_{3\times3}$, otherwise we would have an ambiguity in the optimization of these components. This implies that Γ must be an upper triangular matrix. Thus, we can now define Γ as:

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ 0 & \Gamma_{22} & \Gamma_{23} \\ 0 & 0 & \Gamma_{33} \end{bmatrix}. \tag{3.4}$$

In order to fully understand the role of Γ in the model, we applied different transformation matrices to a previously built synthetic cubic object (see Figure 3.1). Due to the particular symmetry of the cube, results drawn from analysing the effects of the transformation matrices in one of the coordinate axis is easily extended to the other two coordinate axis.

For these synthetic tests we used $\mathtt{R} = \mathtt{I}_{3\times3}$, $\Omega = 0_{3\times3}$, $\Lambda = 0_{3\times3}$, and $\mathtt{S}$ a 9×156 matrix representing the quadratic formulation of the synthetic cubic object. Examples of these experiments can be seen in Figure 3.2.

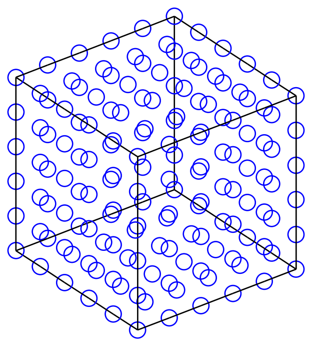


Figure 3.1: Representation of the 3D cubic object used to test the quadratic model for non-rigid bodies. Edges of the object were added to aid visualisation and they are not part of the computations.

From these experiments, we can conclude that the diagonal entries of the linear deformation matrix,

Γ_{11}, Γ_{22} and Γ_{33}, are responsible for the linear expansion/compression of the object, in the direction of the x-axis, y-axis and z-axis respectively (see Figure 3.2 on the left). The off-diagonal entries of Γ (Γ_{12}, Γ_{13} and Γ_{23}) are responsible for pure *shear* deformations (see Figure 3.2 on the right). For instance, let us consider a vector $\mathbf{a}$ initially parallel to the x-axis, and another vector $\mathbf{b}$ initially parallel to the y-axis, and thus orthogonal to $\mathbf{a}$. Γ_{12} is responsible for increasing or decreasing the angle between $\mathbf{a}$ and $\mathbf{b}$, without changing the orientation of $\mathbf{a}$. Thus the vectors will no longer be orthogonal, and the object will present a shear deformation. Analogous results can be drawn for the other entries of the matrix.

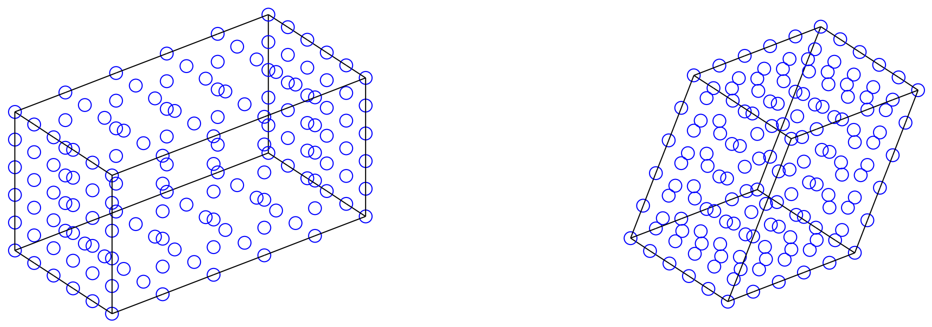


Figure 3.2: On the left, an example of an extension motion on the cubic object caused by $\Gamma_{11} = 1.5$. On the right, an example of the sheer deformation on the cubic object caused by $\Gamma_{13} = 0.5$. The edges on the object were displayed to aid visualisation and they are not part of the computations.

3.1.2 Quadratic deformation and quadratic shape matrices

Following the same procedure as in Section 3.1.1, we examined the properties of the quadratic deformation and quadratic shape matrices by applying different transformations to the already mentioned cubic object (see Figure 3.1). For this case we used $\mathtt{R} = \mathtt{I}_{3\times3}$, $\Gamma = \mathtt{I}_{3\times3}$, $\Delta = 0_{3\times3}$, and $\mathtt{S}$ as the 9×156 shape matrix of the synthetic cube. Examples of the experiments can be seen in Figure 3.3 and Figure 3.4.

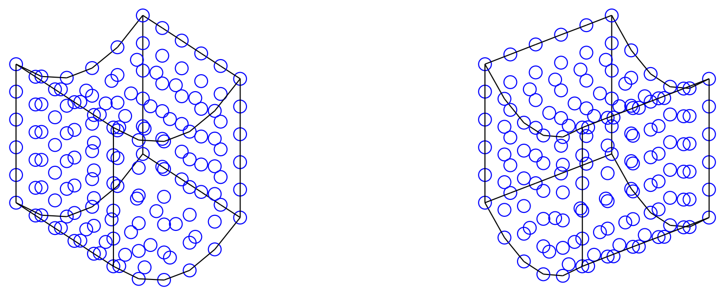


Figure 3.3: On the left, an example of the deformation caused by a non-diagonal entry, with $\Omega_{31} = 0.5$. On the right, an example of the same type of deformation with $\Omega_{32} = 0.5$.

As we can see from Figure 3.3, there are two different bending/bulging possible deformations per

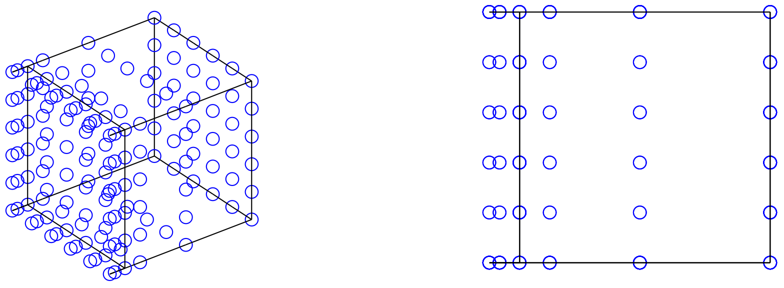


Figure 3.4: On the left, an example of a extension motion on the cubic object caused by $\Omega_{11} = 1$. On the right, a side view of the same example. Note that this deformation has problems as inner planes in the initial shape expand to be outter planes in the final shape.

coordinate axis. These deformations correspond to the off-diagonal entries of Ω (Ω_{12}, Ω_{13}, Ω_{21}, Ω_{23}, Ω_{31} and Ω_{32}) and they can be associated, for instance, to muscular contractions, which are one of the main sources of soft-tissue artifacts. Modelling these deformations is thus very important when dealing with this problem.

However, not every deformation represented by this matrix has a physical counterpart in this particular case. The three diagonal entries of Ω (Ω_{11}, Ω_{22} and Ω_{33}) represent a quadratic extension/contraction along the x-axis, y-axis and z-axis respectively. Still, not only has extension and contraction been (linearly) modelled by Γ, but also this transformation has the problem of plane interpenetration *i.e.* due to its quadratic nature, the relative order among the planes might change during the deformation. This is clearly visible in Figure 3.4, where one of the outermost planes in the initial configuration, where the edge of the cubic object is represented, is now one of the inner planes. Such deformations do not happen when modelling the human body and so, we will not allow these deformations on the model. Finally, we define Ω as:

$$\Omega = \begin{bmatrix} 0 & \Omega_{12} & \Omega_{13} \\ \Omega_{21} & 0 & \Omega_{23} \\ \Omega_{31} & \Omega_{32} & 0 \end{bmatrix}. \tag{3.5}$$

3.1.3 Cross-terms deformation and cross-terms shape matrices

To evaluate the effects of the cross-terms deformation matrix, following the same procedure used in the previous sections. In this case, we will use $\mathtt{R} = \mathtt{I}_{3\times3}$, $\Gamma = \mathtt{I}_{3\times3}$, $\Omega = 0_{3\times3}$ and $\mathtt{S}$, as before, is a 9×156 matrix representing the quadratic formulation of the synthetic cubic object (see Figure 3.1). Examples of the experiments can be found in Figure 3.5 and Figure 3.6.

With this transformation matrix two different kinds of deformations are observed. One of the deformations is characterized by a lateral contraction in one side of the object, while on the opposite side a lateral extension is observed. As we can see from Figure 3.5, this deformation has two modes per axis, and thus we have six matrix entries which gives such deformations: Λ_{11}, Λ_{22}, Λ_{33}, Λ_{13}, Λ_{21} and

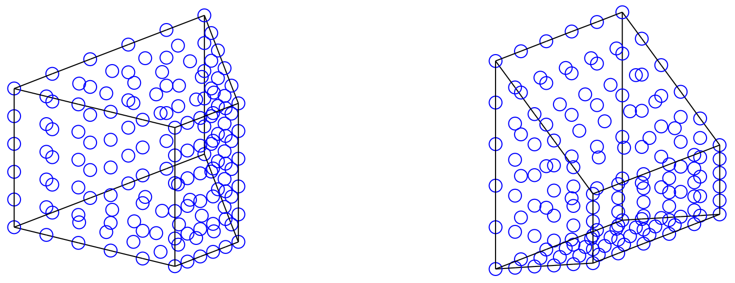


Figure 3.5: Examples of the lateral contraction/extension deformation observed with the cross-values deformation matrix. The deformation on the left corresponds to $\Lambda_{11} = 0.5$, while the one on the right corresponds to $\Lambda_{32} = 0.5$.

Λ_{32}.

The other kind of deformation observed can be described as *twisting* the object around each of the coordinate axis (see Figure 3.6). The matrix entries responsible for this kind of motion are the remaining three entries: Λ_{12}, Λ_{23} and Λ_{31}.

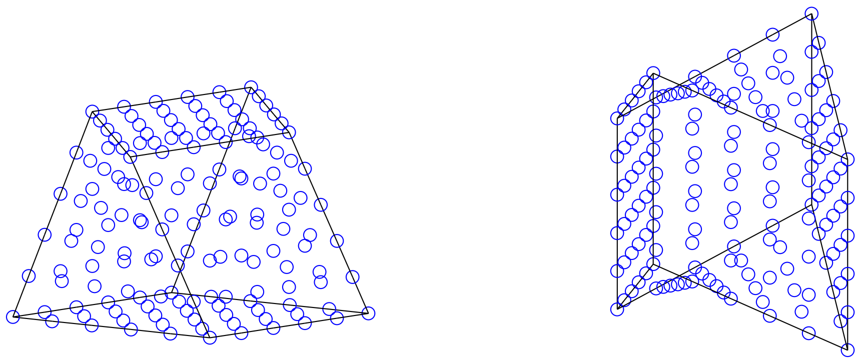


Figure 3.6: Examples of the twisting deformation observed with the cross-values deformation matrix. The deformation on the left corresponds to $\Lambda_{12} = 0.5$, while the one on the right corresponds to $\Lambda_{31} = 0.5$.

On the other hand, the twisting deformation mode is not expected to happen when modelling human body parts, and so it will not be allowed to vary. Summarising, we define the cross-value deformations matrix as:

$$\Lambda = \begin{bmatrix} \Lambda_{11} & 0 & \Lambda_{13} \\ \Lambda_{21} & \Lambda_{22} & 0 \\ 0 & \Lambda_{32} & \Lambda_{33} \end{bmatrix}. \tag{3.6}$$

3.1.4 Model bounds

The deformations allowed by this model have physical meaning only up to a certain degree. For instance, we do not expect a body segment to expand indefinitely, or to contract until all the points collapse on a plane. Similar issues are present in all the deformation modes allowed by the model. Additionally, when applying the model without bounds for these values, even thought the motion reconstruction is accurate, the rigid component of the shape factor will be different from the shape present on the motion (see Figure 3.7). This happens because if any deformation is allowed, the model will generate unrealistic deformations, to which will correspond unrealistic shape factors. Therefore, we must define an upper and lower bound to the entries of the transformation matrices as to prevent meaningless deformations. On the other hand, we do not want the model to become overconstrained, as this would prevent correct modelling of the soft-tissues. Thus, based solely on empirical evaluation of the effects of the bounds on the transformation matrices, we defined the upper and lower bounds as a ± 0.5 from the value of the parameters on the rigid body case. Finally, the upper and lower bounds can be formed as:

$$UB_{\Gamma} = \begin{cases} 1.5, & \text{for diagonal entries.} \\ 0.5, & \text{for off-diagonal entries.} \end{cases};$$

$$LB_{\Gamma} = \begin{cases} 0.5, & \text{for diagonal entries.} \\ -0.5, & \text{for off-diagonal entries.} \end{cases};$$

$$UB_{\Omega} = 0.5;$$

$$LB_{\Omega} = -0.5;$$

$$UB_{\Lambda} = 0.5;$$

$$LB_{\Lambda} = -0.5;$$

where LB stands for lower bound, and UB stands for upper bound.

As a further observation, biological soft-tissues are generally viewed as visco-hyperelastic, incompressible materials [44]. Thus, imposing a volume conservation constraint on our model is of the most importance, as it has a strong physical base. This can be done by requiring the determinant of the Linear deformation matrix to be unitary [44, 43]. This constraint can be written analytically by forcing the following relation:

$$\Gamma_{11}\Gamma_{22}\Gamma_{33} = 1$$

.

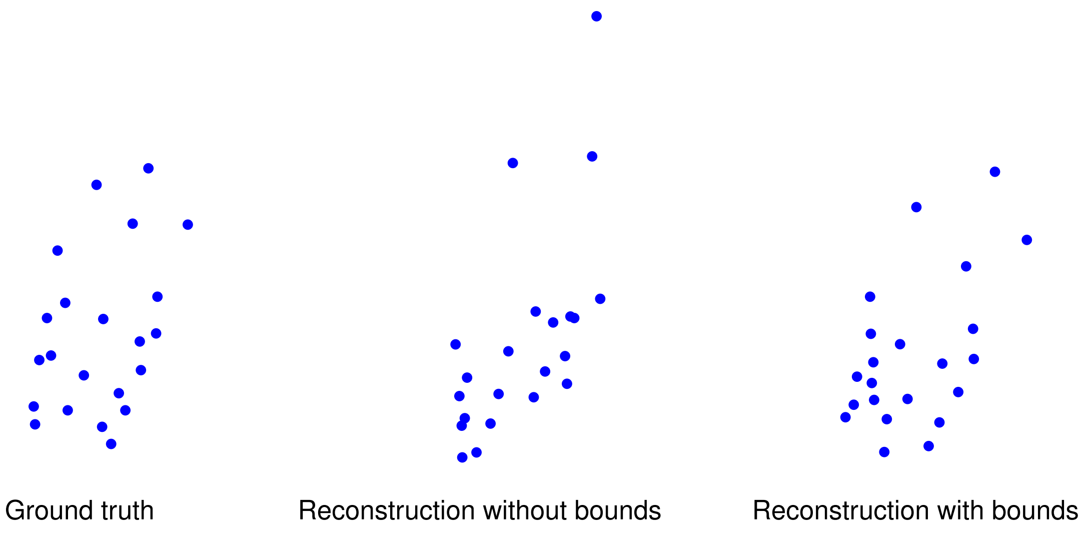


Figure 3.7: On the left, an example of the shape observed on the measurement matrix. On the middle, the rigid component of the shape matrix when using upper and lower bounds on the model. On the right, the rigid component of the shape matrix if no bounds on the deformation are used. It is clear from these images that if no bounds are applied, the rigid shape recovered will be different from the observed shape during the motion.

3.2 Non-linear optimization with a quadratic model

Implementing an iterative method, similar to the one presented in Section 2.3.1, to solve the factorization problem with the *Quadratic model* proved not to be a straight forward task. The reason is mainly because the overall quadratic model is highly non-linear both in the motion and shape components. Thus, adopting a linear alternation scheme as in Section 2.3.1 would lead to a minimisation in which the cost functions are still non-linear at each step. For instance, in the estimation of the motion parameters we have a rotation matrix which multiplies the quadratic deformation parameters. Similarly, in the shape components, we have the squared and cross-product version of the 3D coordinates. For these reasons, we decided to directly adopt a non-linear optimization approach, initialized by the rigid-body motion estimated by our weighted factorization approach described in Section 2.3. We defined a non-rigid cost function which will then be optimized using a Levenberg-Marquardt iterative optimization scheme, generically called bundle-adjustment (BA).

3.2.1 The non-rigid cost function

Our goal is to find the best representation of the data matrix $\mathtt{W}$ by using the quadratic model described in Section 3.1. When reconstructing the data matrix $\mathtt{W}$ based on a model, there will always be some residual error associated to it caused, for instance, by the noise in the data, computational errors or model inaccuracies. The 3-vector of the 3D coordinates estimated by our model for point j at frame i can thus be represented as:

$$\tilde{\mathbf{w}}_{ij}^{(rec)} = \mathtt{M}_i \mathbf{s}_j = \mathtt{R}_i \left[\Gamma_i \, \Omega_i \, \Lambda_i \right] \mathbf{s}_j ; \tag{3.7}$$

while the residual error of reconstruction can be defined as:

$$\mathbf{e}_{ij} = \tilde{\mathbf{w}}_{ij} - \tilde{\mathbf{w}}_{ij}^{(rec)}, \tag{3.8}$$

where $\tilde{\mathbf{w}}_{ij}$ is the data acquired by the MOCAP system.

The best fit of the quadratic model to the MOCAP data is then found by minimizing the norm of the residual errors such that:

$$\underset{\mathtt{R}_i, \Gamma_i, \Omega_i, \Lambda_i, \mathbf{s}_j}{\arg\min} \sum_{i,j}^{F,P} ||\mathbf{e}_{ij}||^2 = \underset{\mathtt{R}_i, \Gamma_i, \Omega_i, \Lambda_i, \mathbf{s}_j}{\arg\min} \sum_{i,j}^{F,P} ||\mathbf{w}_{ij} - \tilde{\mathbf{w}}_{ij}||^2 . \tag{3.9}$$

However this sum of non-linear cost function has two major disadvantages. The number of parameters to be estimated rises considerably with the complexity of the object to be modelled, making the computational cost too big for this method to be feasible. Also, the combination of several parameters can create multiple local minima, resulting in difficult convergence to the global minimum. To solve this problem, we propose an optimization procedure that takes advantage of the particular properties of this problem.

3.2.2 The bundle-adjustment minimisation approach

BA refers to a combination of techniques which sum up to an efficient scheme for minimisation (*i.e.* Levenberg-Marquardt) and computational tools which lower the computational requirements of the method. Levenberg-Marquardt methods use a combination of Gauss-Newton and gradient descent minimisation schemes, alternating from one to the other whenever the conditions are favourable and they have been extensively tested in may engineering applications such as photogrammetry [45] and computer vision [46]. However, the computation load is increasing dramatically when dealing with big parameter space such as the one given by the quadratic model. The major contribution to the computational burden of these second-order algorithms is represented by the computation of the inverse matrix of the Hessian of the cost function, in every iteration of the Gauss-Newton descent step.

Let us define a N-vector Θ containing all the parameters to be estimated as:

$$\Theta = \left[\Theta_{\mathtt{R}_1}, \ldots, \Theta_{\mathtt{R}_F}, \Theta_{\mathbf{\Gamma}_1}, \ldots, \Theta_{\Gamma_F}, \Theta_{\Omega_1}, \ldots, \Theta_{\Omega_F}, \Theta_{\Lambda_1}, \ldots, \Theta_{\Lambda_F}, \Theta_{\mathbf{s}_1}, \ldots, \Theta_{\mathbf{s}_P} \right]^T , \tag{3.10}$$

where $\Theta_{\mathtt{R}_i}$, Θ_{Γ_i} , Θ_{Ω_i}, Θ_{Λ_i} and $\Theta_{\mathbf{s}_j}$ represent respectively the parameters for the rotation, linear deformations, quadratic deformations, cross-term deformations and shape matrix parameters for all the frames and points. Based on these definitions, the cost function $\mathtt{K}$ can be defined as:

$$\mathtt{K}(\Theta) = \sum_{i,j}^{F,P} ||\mathbf{e}_{ij}||^2 . \tag{3.11}$$

The Gauss-Newton minimisation approach is an iterative algorithm where at each step t an update Δ^t is added to the previous solution:

$$\Theta^{t+1} = \Theta^t + \Delta^t, \tag{3.12}$$

in such a way it decreases the sum of the residual errors in $\mathtt{K}$. Let us define a $3FP$-vector $\mathbf{e}$ containing all the residuals for each frame i and point j such that $\mathbf{e} = \left[\mathbf{e}_{11}^T, \ldots, \mathbf{e}_{FP}^T\right]^T$. Dropping the index t for the sake of notation simplicity, if we assume local linearities in the cost function, we can now expand equation (3.12) as a second order Taylor series as:

$$\mathtt{K}(\Theta + \Delta) \approx \mathtt{K}(\Theta) + \mathbf{g}^T\Delta + \frac{1}{2}\Delta^T\mathtt{H}\Delta, \tag{3.13}$$

where the N-vector $\mathbf{g} = \mathtt{J}^T\mathbf{e}$ is the gradient vector, with $\mathtt{J} = \dfrac{\partial \mathbf{e}}{\partial \boldsymbol{\Delta}}$ being the $3FP \times N$ jacobian matrix of the model parameters, and $\mathtt{H}$ being the $N \times N$ Hessian matrix of the cost function, which can be approximated as $\mathtt{H} = \mathtt{J}^T\mathtt{J}$ (for more details see [46] about the Gauss-Newton approximation of the Hessian matrix).

The increment Δ is found by computing $\dfrac{\partial q}{\partial \Delta} = 0$, where $q = \mathbf{g}^T\Delta + \dfrac{1}{2}\Delta^T\mathtt{H}\Delta$. Finally, the Gauss-Newton descent step can be defined as:

$$\mathtt{H}\Delta = -\mathbf{g}. \tag{3.14}$$

The Levenberg-Marquardt optimization method differs from the simple Gauss-Newton method as it introduces a damping term to equation (3.14), resulting in:

$$(\mathtt{H} + \lambda\mathtt{I}_{N\times N})\,\Delta = -\mathbf{g}, \tag{3.15}$$

where $\mathtt{I}_{N\times N}$ is the $N \times N$ identity matrix. This additional damping term allows controlling the algorithm convergence in order to switch back and forth from a Gradient Descent to Gauss-Newton methods, in order to better fit the problem conditions. This also guarantees numeric stability by forcing $\mathtt{H} + \lambda\mathtt{I}_{N\times N}$ to be a full-rank matrix, and thus invertible.

Solving the set of equations defined by equation (3.15) is problem of complexity $O(N^3)$ and has to be done at every iteration, making the computational cost of this problem too high when the number of parameters increase. Still, we can use the properties of the factorization problem to find more efficient ways to deal with this problem. If we consider the motion and shape parameters $(\mathtt{R}_i, \Gamma_i, \Omega_i, \Lambda_i)$, they are completely independent among each other on every frame. Similarly, the P shape parameters $\mathbf{s}_j$ are independent among each other. Thus the Jacobian matrix J will have a very sparse structure.

Since the Hessian matrix H is computed based on this Jacobian, it will also have a sparse structure.

By using standard approaches for sparse matrix computation, the complexity of the problem of inverting H can be reduced [47, 48]. With this, applying bundle-adjustment to the non-rigid body factorization becomes feasible even for a large number of parameters.

Chapter 4

MATLAB Analysis Tool

For segmenting the set of points and visualize the results, we developed two MATLAB-based software tools. We emphasise that it is not our goal to create a very sophisticated software for extensive manipulation of the data. Instead our goal is to create a simple software tool that allows an easy manual segmentation of the data to use in our algorithms. Although numeric information is very important to assess the validity of the algorithm, the ability to visualise the captured data and the generated model are essential to fully understand the capabilities of these algorithms. Thus we also provide a software tool that allows interactive visualisation of the data.

4.1 Segmentation software tool

The general layout of the segmentation software tool can be seen in Figure 4.1.

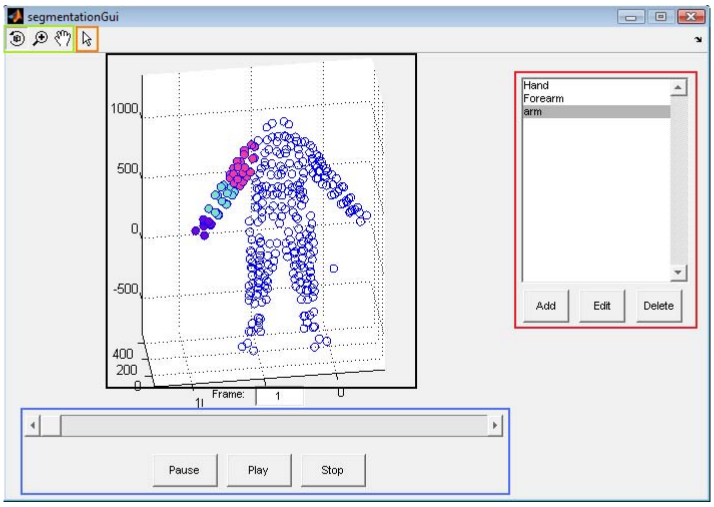
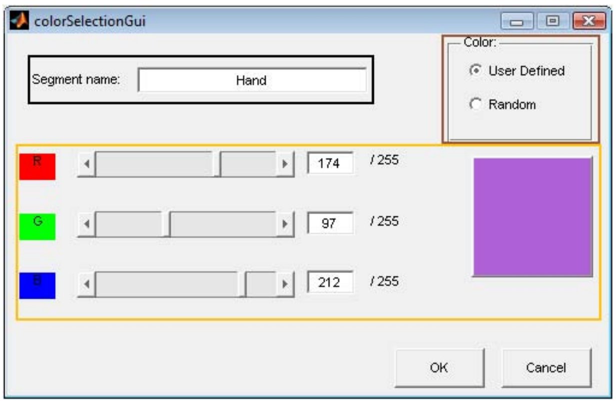


Figure 4.1: General layout of the Segmentation Tool. The plot window is highlighted in black. The segment editor options are highlighted in red. The animation player options are highlighted in blue. The camera options are highlighted in green. Highlighted in orange is the *marker selection* option

.

The goal of this software tool is to allow the user to create as many segments as needed, and to manually select the markers that belong to each one of them. The 3D coordinates of the markers, provided by the MOCAP system, are displayed on the plot window as blue circles (see Figure 4.1, highlighted in black). The software tool has a player-like environment allowing the user to visualise the data frame by frame or play it in *video mode* (see Figure 4.1, highlighted in blue). To take full advantage of the 3D data, it is also possible to *change the viewpoint*, *zoom* in or out and *panning* the image, even when playing in video mode (see Figure 4.1, highlighted in green).

Segments are defined by a name and a color. When adding a segment to the segment list (see Figure 4.1 highlighted in red), an additional window appears to define these properties. An example of such a window can be seen in Figure 4.2. The color assigned to the segment is chosen by defining an RGB code. The color defined by the current RGB code is also displayed on the window (see Figure 4.1 highlighted in orange). Segments can be edited or deleted whenever the player is paused or stopped.


colorSelectionGui

Segment name: Hand

Color: [✓ User Defined] Random

R 174 / 255

G 97 / 255

B 212 / 255

Figure 4.2: Dialog for defining the properties of the segments.

Assigning markers to a segment is done by using the *Select Point* tool (see Figure 4.1, highlighted in orange). Clicking on a marker will then assign it to the segment highlighted on the list, while its color on the animation will change to the color defined for the segment. Examples of selected markers can be seen in Figure 4.1, represented by markers in three different colors, belonging to the segments for the hand, forearm and upper arm. Clicking on a selected marker will remove it from the previously assigned segment. This procedure is not available in video mode, it must be done while the player is paused or stopped.

4.2 Visualization software tool

The visualization software tool has a similar appearance to the segmentation software tool described above. A general layout of the software tool can be seen in Figure 4.3. This tool uses the same player-like environment to display an animation of the output of the MOCAP systems and the data resulting from our algorithms (see Figure 4.3, highlighted in black). The options for *zooming*, *panning* and *rotating* the images are also available (see Figure 4.3, highlighted in blue). However the purpose of this tool is to visualize the different outputs from our algorithms, and so the functionalities available are quite different.

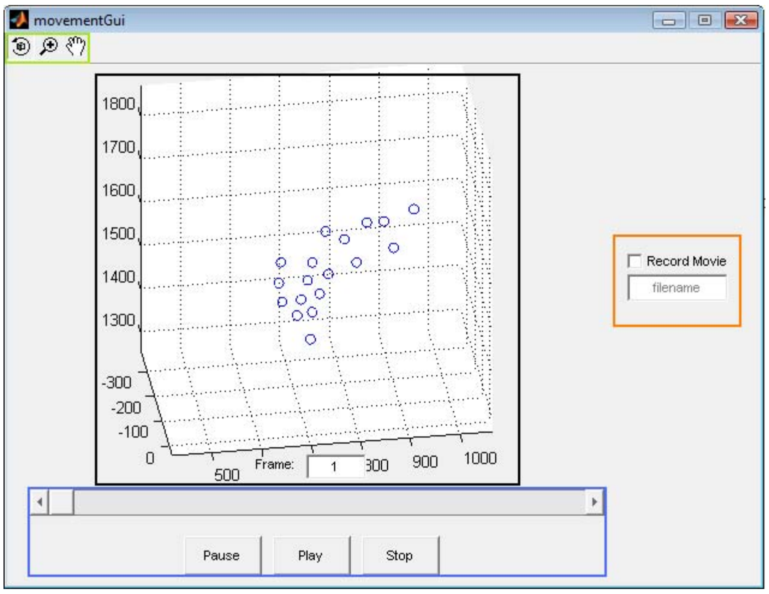


Figure 4.3: General layout of the Visualization Tool. The plot window is highlighted in black. The animation player options are highlighted in blue. The camera options are highlighted in green. The check box option for exporting the animation as a video file is highlighted in orange.

The tool allows us to visualize two data matrices simultaneously. This option was created in order to visually compare the accuracy of the reconstructions resulting from our algorithms. In this mode, one of the data sets is displayed as blue circles while the other is displayed as red asterisks (see Figure 4.4).

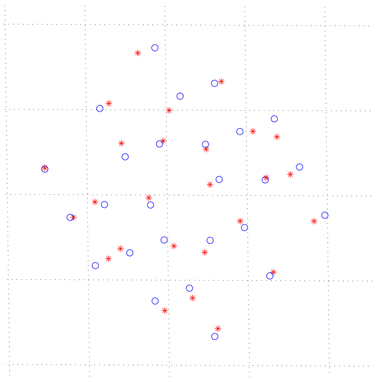


Figure 4.4: Example of the Visualization Tool displaying two shapes in motion at the same time. One of the shapes is displayed using blue circles and the other is displayed as red asterisks.

For a better perception of the results for the joint parameters, this tool can also display representations of the reconstructed joints. The joint centres recovered for the universal joint are displayed as a combination of a red circle and a red asterisk. This symbol is also displayed larger then circles used for the markers, as to stand out in the point clouds (see Figure 4.5). For the hinge joints, we represent the axis of rotation using a green line built using equation (2.29) (see Figure 4.6).

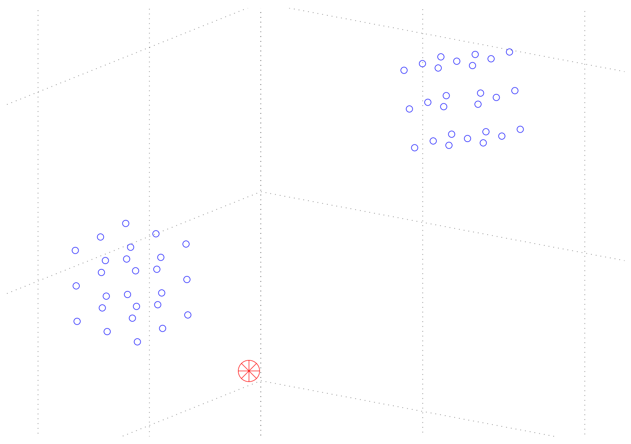


Figure 4.5: Example of the Visualization Tool displaying a universal joint. The markers of the two bodies are displayed as blue circles while the joint centre is displayed as a red circle combined with a red asterisk.

There is also an option to export the animation to a video file (see Figure 4.3). However, when doing so, the animation has to played from beginning to end, without manipulating the playback or camera controls.

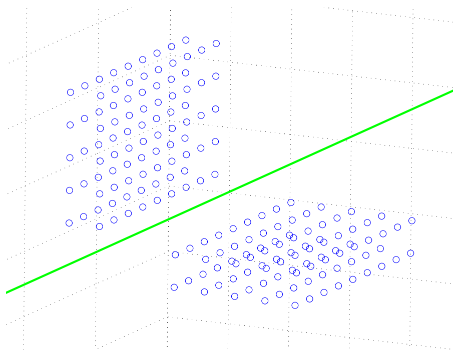


Figure 4.6: Example of the Visualization Tool displaying a hinge joint. The markers of the two bodies are displayed as blue circles while the joint axis is displayed as a green line.

Chapter 5

Experimental Results

The algorithms studied in Chapter 2 and Chapter 3 were implemented in MATLAB, a high level language developed for numerical computations, matrix operations and graphic visualisation.

For the experimental validation of our algorithms, we generated synthetic data specifically for that task. As determining a joint's centre or axis on a human subject is not trivial, the synthetic data is able to provide us with easily accessible ground truth data for our experiments. Afterwards, we applied our algorithms to real MOCAP output data taken from the database of the Graphic Lab of Carnegie-Mellon University (http://mocap.cs.cmu.edu/) and also MOCAP data acquired at the Augmented Human Interaction Laboratory of the Department of Computer Science of Queen Mary University of London. In both cases the data was captured using a VICON commercial MOCAP system. VICON is an active optical system that works with infrared light. The light emitted by the system is then reflected on the markers, and captured by infrared sensors. With this information the position of the markers can be easily computed.

Since we do not know the ground truth measurements for these cases, we present the resulting animation in order to have a qualitative evaluation of the joint properties. We chose not only movements that are able to exemplify each of the joint types used to model the articulations, but also very common movements, mainly from sports, that would give real-life examples of potential applications of these algorithms. Notice that by using different databases we can show that these algorithms do not depend on the MOCAP system's setup. Since the available data is not segmented by default, for the experimentation the data was hand segmented and visualized using the MATLAB software tools presented on Chapter 4.

5.1 Weighted factorization

The weighted factorization algorithm is a focal step in the new proposed approach, as it reduces noise it and estimates a more accurate rigid shape representation. To test its performance, we built a synthetic test representing a cubic object, of side 2 units, containing 26 feature points and performing random motions on a scene. We tested the algorithm with different levels of *additive white Gaussian noise* (AWGN), and later performed a statistical analysis of the accuracy of the reconstruction. Finally, we applied the algorithm to real data obtained by a Vicon MOCAP.

5.1.1 Performance measurements

The cubic object used in the synthetic data is represented in Figure 5.1. Edges of the object are displayed just to aid visualisation and are not part of the computations. The synthetic feature points are represented in red. Based on this object we built several random motions that were used as a test battery for our algorithm.

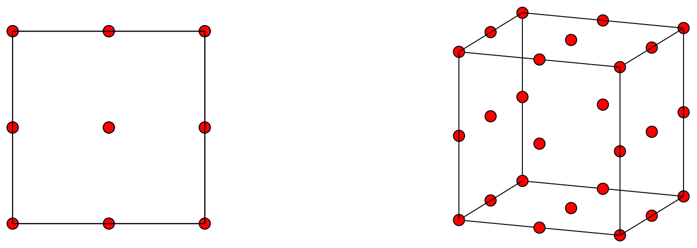


Figure 5.1: MATLAB plots of the cubic object used for the synthetic tests. The synthetic feature points are represented as red dots. The edges are shown to aid visualisation and they are not included in computations.

Performances were measured based not only on the accuracy of the reconstruction of the shape matrix, but also on the overall result on the data matrix. However, we cannot make a straight forward comparison between the ground truth shape and the reconstructed shape. As we saw in Chapter 2, the factorization problem has an ambiguity in the solution. The obtained solution is always valid up to an unknown rotation $\mathtt{R}_P$ since:

$$\mathtt{W} = \mathtt{MS} = \mathtt{MR}_P\,\mathtt{R}_P^T\mathtt{S} = \tilde{\mathtt{M}}\tilde{\mathtt{S}}. \tag{5.1}$$

In practice this is not a problem as there is no such thing as a real shape matrix $\mathtt{S}$. However, when we want to compare ground truth data to reconstructed data, the ground truth shape and the reconstructed shape have to be previously registered to avoid this ambiguity. This is done by performing a Procrustes analysis on the data [41]. Let us define $\mathtt{S}^{(gt)}$ and $\mathtt{S}^{(rec)}$ as respectively the ground truth and reconstructed shape matrices. Registering the shape matrices using the Procrustes analysis consists of solving the least-squares problem defined by:

$$\arg\min_{\mathtt{R}^{(P)}} \left|\left|\mathtt{S}^{(rec)} - \mathtt{R}_P\,\mathtt{S}^{(gt)}\right|\right|, \tag{5.2}$$

with $\mathtt{R}_P$ constrained to be a rotation matrix (for more details see Appendix B).

Another factor that prevents straight forward comparison between $\mathtt{S}^{(gt)}$ and $\mathtt{S}^{(rec)}$ is the scale of the objects. This difference may result from the noisy data provided as an input. To solve that, let us define

l_{gt} as the length of the edge of the ground truth cubic object. This value is a useful parameter to use with cubic objects as all the edges have equal length. On the other hand, the reconstructed object might not be exactly cubic and so the edges cannot be assumed to be equal. Let us define l_r as the mean value of the length of all the edges of the reconstructed object. The scaling factor can now be defined as:

$$l_s = \frac{l_{gt}}{l_r}. \tag{5.3}$$

Let us also define a residual error matrix $\mathtt{E}^{(S)}$ as:

$$\mathtt{E}^{(S)} = \mathtt{S}^{(gt)} - \mathtt{R}^{(P)}\,\mathtt{S}^{(rec)}. \tag{5.4}$$

This matrix represents the differences between the parameters of the ground truth and reconstructed shape matrices once the rotation ambiguity has been removed. We can now use equations (5.3) and (5.4) to define the measurement of the root means squared (RMS) error for the global shape matrix as:

$$\varepsilon^{(S)} = \sqrt{\frac{1}{P \times l_s} \sum_{j=1}^{P} \left|\left|\mathbf{E}_j^{(S)}\right|\right|^2}, \tag{5.5}$$

where $\left|\left|\mathbf{E}_j^{(S)}\right|\right|$ represents the error on the global shape for point j. The factor $\frac{1}{P}$ is a normalising factor for the number of points in the shape.

However, this error measurement is still system specific. In order to have system independent error measurement, some kind of normalisation must be done. For this purpose, we chose to normalise the RMS error by the norm of the ground truth shape matrix. We finally define the error measurement as:

$$\varepsilon_n^{(S)} = \frac{\varepsilon^{(S)}}{||\mathtt{S}^{(gt)}||}. \tag{5.6}$$

We defined similarly the error for the data matrix as:

$$\mathtt{E}^{(W)} = \mathtt{W}^{(gt)} - \mathtt{W}^{(rec)}, \tag{5.7}$$

with $\mathtt{W}^{(gt)}$ the ground truth data matrix and $\mathtt{W}^{(rec)}$ the reconstructed data matrix. Note that this matrix is different from the error matrix $\mathtt{E}$ presented in Section 2.3.1 as that matrix was built using a different arrangement for the 3D coordinates in the data matrix.

There is no need for a Procrustes analysis in this case, as there is no ambiguity in recovering $\mathtt{W}^{(rec)}$. The reconstructed matrix is the best approximation the algorithm can return for $\mathtt{W}^{(gt)}$. However, in this case, we must not only normalise for the number of feature points P, but also for the number of frames F. The RMS error for the global data matrix can thus be defined as:

$$\varepsilon^{(W)} = \sqrt{\frac{1}{P \times F} \sum_{i=1}^{F} \sum_{j=1}^{P} \left|\left|\mathbf{E}_{ij}^{(W)}\right|\right|^2}, \tag{5.8}$$

where $\left|\left|\mathbf{E}_{ij}^{(W)}\right|\right|$ is the residual error for the feature point j at frame i in the global data matrix. Again, the RMS error is normalised by $||\mathtt{S}^{(gt)}||$:

$$\varepsilon_n^{(W)} = \frac{\varepsilon^{(W)}}{||\mathtt{S}^{(gt)}||}. \tag{5.9}$$

5.1.2 Weighted factorization with additive Gaussian noise

Now that we have defined how to compute the error for the algorithm, we can carry out an analysis of its performance. Using the test battery we created, we tested the accuracy of retrieving the shape matrix and reconstructing the original motion, defined respectively by equations (5.6) and (5.9), on 1000 completely random motions using 7 different levels of AWGN. The noise levels were defined based on the variance of the Gaussian distribution, and had the following values: $\sigma^2 = 0$, 0.01, 0.05, 0.1, 0.2, 0.4, and 0.6. In order to avoid the algorithm being stuck on a local minima, we limited the iterations to 500 on every test. To compare the performance, we also submitted the non-weighted factorization approach presented in Section 2.1 to the same tests. The statistical analysis of the data is represented by box plots (*e.g.* Figure 5.2). A box plot consists of a blue box for each test condition (in this case, noise level), delimited by the first quartile (χ_{25}) at the bottom, and the third quartile (χ_{75}) at the top. The red line in the middle of the box represents the median (χ_{50}) of the samples. The range of the data is determined by the interquartile range (IQR) defined by $IQR = \chi_{75} - \chi_{25}$. All data that lies higher then $\chi_{75} + 1.5 \times IQR$, or lower then $\chi_{25} - 1.5 \times IQR$ is considered an outlier. When existent, these are represented by a red plus sign. Black dashed lines extend from the top and bottom of the blue box until the last non-outlier value. The box plot analysis of the error for the shape matrix reconstruction is represented in Figure 5.2. The mean error values versus the noise levels for the shape and data matrix reconstructions can be found in Table 5.1.

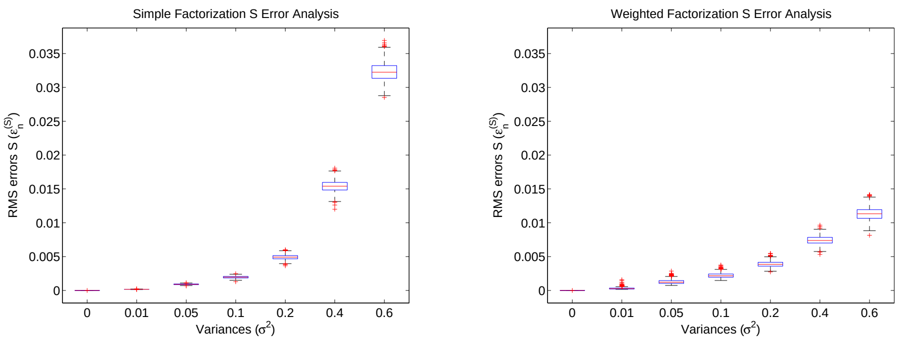


Figure 5.2: On the left, the box plot for the analysis of the shape matrix reconstruction using the simple factorization method. On the right the box plot for the analysis of the shape matrix reconstruction using the weighted factorization method.

Figure 5.2 and on Table 5.1 show that while both algorithms, weighted and non-weighted, work perfectly in the recovery of the shape matrix in the ideal case ($\sigma^2 = 0$), their performances are different

Table 5.1: Mean Error values for Shape and Data Matrix reconstruction using Simple and Weighted Factorization algorithms

	Simple Factorization		Weighted Factorization	
σ^2	$\varepsilon_n^{(S)}$ (%)	$\varepsilon_n^{(W)}$ (%)	$\varepsilon_n^{(S)}$ (%)	$\varepsilon_n^{(W)}$ (%)
0	2.24×10^{-13}	1.31×10^{-4}	5.16×10^{-14}	3.16×10^{-15}
0.01	0.017	0.00057	0.027	0.00040
0.05	0.090	0.028	0.120	0.020
0.1	0.197	0.056	0.217	0.031
0.2	0.491	0.112	0.387	0.076
0.4	1.54	0.229	0.738	0.159
0.6	3.23	0.333	1.13	0.232

when the data is noisy. When the variance of the AWGN is small ($\sigma^2 = 0.01$ and 0.05) , the non-weighted algorithm performs better then our weighted algorithm. The number of outliers is also higher on the weighted algorithm, suggesting that in those cases the algorithm may be trapped in local minima. Still, the performance of our weighted algorithm matches the non-weighted algorithm approximately when $\sigma^2 = 0.1$, surpassing it for higher noise levels ($\sigma^2 = 0.2$, 0.4 and 0.6), both in median error and in number of outliers.

The box plot analysis of the reconstructed data matrix is presented in Figure 5.3. In this case, for small levels of noise ($\sigma^2 = 0, 0.01, 0.05$ and 0.1) the weighted and non-weighted algorithms have similar performances. Still, for higher noise levels ($\sigma^2 = 0.2$, 0.4 and 0.6) the weighted algorithm has, again, a better performance.

The difference between the performances with noisy data is not as significant for the data matrix as it is for the shape matrix case. This can be explained by the fact that, when applying the orthogonality constraints to the motion matrix, the simple factorization algorithm finds a global solution for $\mathtt{M}$ (see equation 2.13). When data is noisy, it is likely that in many frames $\mathtt{M}_i$ is not exactly a rotation matrix, thus allowing affine transformations of the shape matrix. Without enough constraints on the motion matrix, the shape matrix will not be accurately estimated. On the other hand, the weighted factorization algorithm requires every matrix $\mathtt{M}_i$ to be a rotation matrix, at the expense of disregarding the continuity of the motion. Still, when higher levels of noise are used, this algorithm can not only provide a more accurate estimation of $\mathtt{W}$, but also a more accurate estimation of $\mathtt{S}$.

Since we are aiming to deal with deformable objects, the weighted factorization algorithm seems to be more suited for recovering a rigid structure based on the motion of a deformable object. Note that recovering accurate rigid structure and motion matrices is important for the initialization of our quadratic model for non-rigid bodies, presented in Chapter 3.

5.1.3 Weighted factorization with occlusion

Another feature of our Weighted Factorization algorithm as presented in Section 2.3.3 is the ability to handle occlusion. Still, it is important to quantify the amount of missing data the algorithm can

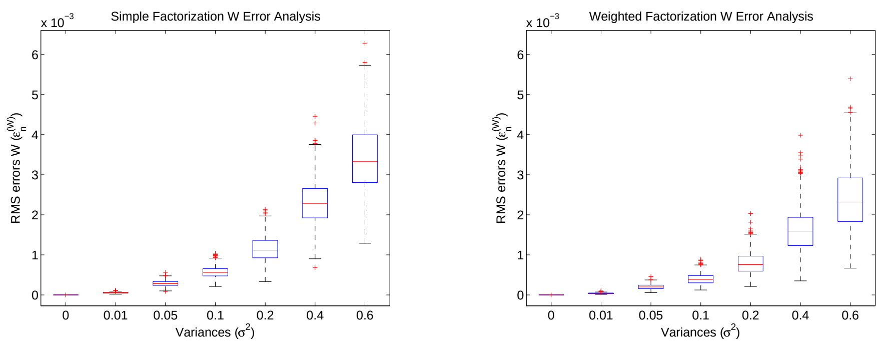


Figure 5.3: On the left, the box plot for the analysis of the data matrix reconstruction using the simple factorization method. On the right the box plot for the analysis of the data matrix reconstruction using the weighted factorization method.

handle in various conditions. For this purpose we generated the Z matrix defined in Section 2.3.3 to have different percentages of missing entries: 30%, 40%, 50% and 60%. Since these missing entries are assigned randomly, for each different percentage we applied the algorithm to the 100 random rigid body motions existent in our test battery. To test the robustness of the algorithm, we tested each case with 7 different levels of AWGN, defined by the value of its variance: $\sigma^2 = 0, 0.01, 0.05, 0.2, 0.4$ and 0.6. Performance was again measured using the same normalised RMS error defined in equation (5.6).

In Figure 5.4 we present the error analysis for the shape matrix reconstruction with occlusion. Data regarding the mean error value for each noise level and missing data percentage, as well as the number of non-convergent cases, given by N_Z, can be seen in Table 5.2.

Table 5.2: Mean error for the shape matrix S with different levels of noise and missing data.

	$p = 0.3$		$p = 0.4$		$p = 0.5$		$p = 0.6$	
σ^2	$\varepsilon_n^{(S)}$	N_Z	$\varepsilon_n^{(S)}$	N_Z	$\varepsilon_n^{(S)}$	N_Z	$\varepsilon_n^{(S)}$	N_Z
0	5.04×10^{-14}	0	4.62×10^{-14}	0	4.35×10^{-14}	3	0.350	53
0.01	0.0398	0	0.0382	0	0.0410	0	0.0748	34
0.05	0.142	0	0.158	0	0.176	0	0.218	41
0.1	0.254	0	0.273	0	0.301	1	0.381	45
0.2	0.469	0	0.502	0	0.554	0	0.646	53
0.4	0.897	0	0.968	0	1.05	1	1.24	48
0.6	1.38	0	1.55	0	1.67	4	1.89	47

From these results we can see that our algorithm is able to successfully deal with 50% of missing data. Still, due to the randomness of the occlusion phenomenon, it is possible that the algorithm will not converge even when only 30% of the data is missing. This can happen if, for instance, missing data is particularly strong on a given frame, making impossible to solve the system of equations for that case. With the levels of occlusion tested, the algorithm only failed to converge in 9 cases when occlusion was at most 50%, all of them when using the highest level of noise. When dealing with 60% of missing data, approximately half of the tests will fail. Thus we consider our algorithm cannot handle 60% of missing data. The true limit value will lie between 50% and 60%. Still, as no more tests were done and

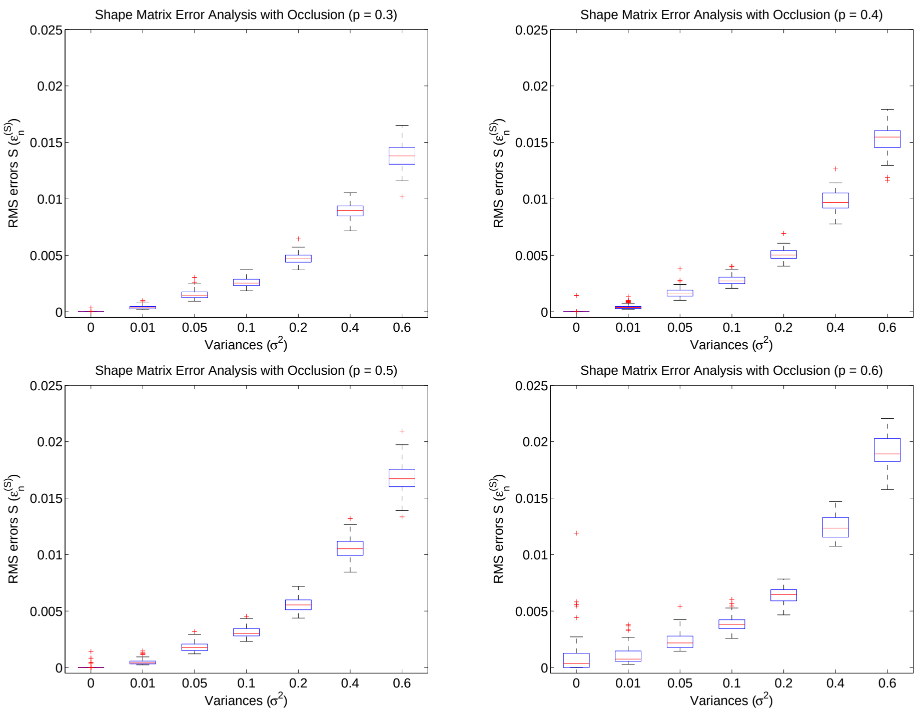


Figure 5.4: Box plot analysis of the reconstruction of the shape matrix, with 30%, 40%, 50% and 60% of occlusion.

also to have a safety margin, we will consider our algorithm to be limited to 50% missing data. When comparing data with occlusion and data without occlusion (see Section 5.1.2) we can see that when no noise is added, even with 50% of occlusion, we have a very a similar performance. However, as the error level increases, the reconstruction based on missing data will be getting, as expected, slightly worse. Within the same level of noise, the performance also decreases when more data is occluded.

5.2 Universal joint

Following a similar approach as used in Section 5.1 we first tested our universal joint factorization algorithm using synthetic data. This data simulated two cubic rigid objects, composed of 26 feature points each, linked by an universal joint. With these settings a test battery of 1000 completely random motions over 500 frames of that object was created. The cubic rigid objects used where the same defined in Section 5.1. The universal joint setup can be seen in Figure 5.5.

To test the robustness of the universal joint factorization algorithm, we ran the test battery with different levels of AWGN.

5.2.1 Performance measurements

To be able to compare performances we must first define an error measurement for the algorithm. When using the universal joint factorization defined in Section 5.2, $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$ are the parameters

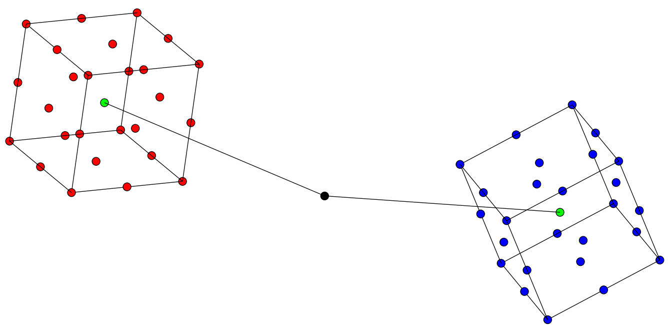


Figure 5.5: A MATLAB plot of the synthetic setup of the universal joint. The first body's feature points are represented as blue dots. The second body's feature points are represented as red dots. The object centroids are represented as green dots. The vectors $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$ are represented as a black line. The joint centre is represented as a black dot. The edges of the objects represented as black lines were added to facilitate visualisation and they are not part of the computations

that define the joint. These vectors are computed based on the shape and motion parameters of the two bodies, thus their estimate reflects the accuracy of the algorithm on the overall joint. Since the joint axis was defined in equation 2.29 based on $\mathbf{d}^{(1)}$, we will focus on this vector to estimate the accuracy of the algorithm. We already know that $\mathbf{d}^{(1)}$ must fulfil the motion equations of the first body. Thus it is possible to define a data matrix containing the 3D coordinates for the joint centre over all the frames as:

$$\mathtt{W}^{(JC)} = \mathtt{M}^{(1)}\mathbf{d}^{(1)} + \mathbf{t}^{(1)}. \tag{5.10}$$

Let $\mathtt{W}^{(JC_{gt})}$ be the data matrix for the joint centre in the ground truth data, and $\mathtt{W}^{(JC_r)}$ be the data matrix for the joint centre reconstructed by the factorisation algorithm. We can now define an error matrix for the universal joint as:

$$E^{(JC)} = W^{(JC_{gt})} - W^{(JC_r)}. \tag{5.11}$$

Using a similar formulation as the one used in Section 5.1 for the weighted factorization algorithm, we can now define the RMS error for the universal joint algorithm as:

$$\varepsilon^{(U)} = \sqrt{\frac{1}{F}\sum_{i=1}^{F}\left|\left|\mathbf{E}_i^{(JC)}\right|\right|^2}, \tag{5.12}$$

where $\left|\left|\mathbf{E}_i^{(JC)}\right|\right|$ is the residual error for the joint centre at frame i. Since we are only dealing with the coordinates of the joint centre, we only need to normalise regarding the number of frames F.

A system independent value can be found by normalising $\varepsilon^{(U)}$ with the average "size" of the objects. In this case we will normalise with $||\mathtt{S}^{(gt)}||$, where $\mathtt{S}^{(gt)}||$ is the ground truth shape matrix of the cube used on the data, as both objects are equal. Thus, we define the normalised error measurement as:

$$\varepsilon_n^{(U)} = \frac{\varepsilon^{(U)}}{||\mathrm{S}^{(gt)}||} \tag{5.13}$$

5.2.2 Universal joint factorization with additive Gaussian noise

By using equation 5.13 we can now carry out a statistical analysis of the performance of the algorithm. This is done by running the universal joint factorization algorithm to the test battery composed of 1000 completely random universal joint motions using 7 different levels of AWGN. Again, the noise levels are defined based on the variance of the Gaussian distribution, with values: $\sigma^2 = 0$, 0.01, 0.05, 0.1, 0.2, 0.4 and 0.6. The statistical results are represented by a box plot on Figure 5.6. The mean value of the RMS error versus the noise level is presented in Table 5.3.

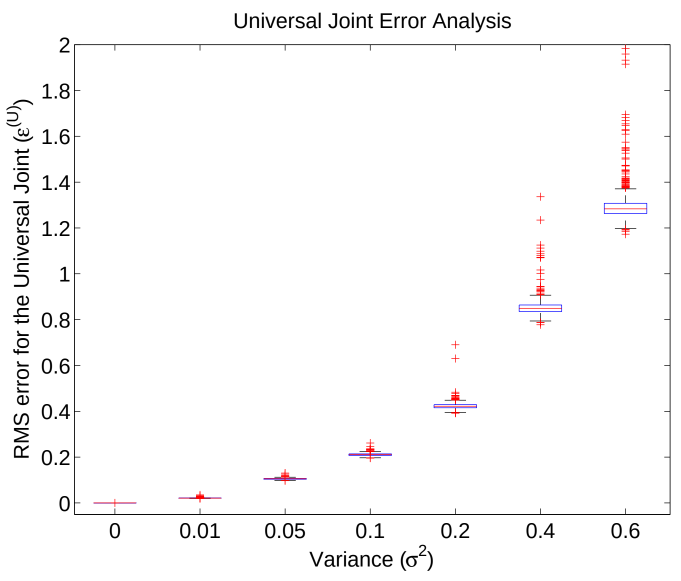


Figure 5.6: Box plot of the error analysis for the Universal Joint centre, with 7 different levels of noise.

Based on Figure 5.6, we can conclude that while the IQR is small for all the cases,indicating a good reliability of the results, there is a relatively high number of outliers. This number is specially high when dealing with higher noise levels. As we are estimating a single point in the 3D space, it seems plausible that higher noise values can have a bigger impact on the computations. In fact, when dealing with higher noise values, the error can even reach 30%, which is unsatisfactory. Still, such error values are not commonly encountered when dealing with real data, being the values presented for realistic errors satisfactory.

5.2.3 Universal joint factorization with real data

To demonstrate the application on real data, we applied the universal factorization algorithm to data captured in the Augmented Human Interaction Laboratory of the Department of Computer Science of Queen Mary University of London. The data analysed in this section consists of two bodies, a

Table 5.3: Mean Error for the Universal Joint Case vs. Noise Level

σ^2	$\varepsilon_n^{(U)}$ (%)
0	$7,47 \times 10^{-13}$
0.01	0.50
0.05	2.50
0.1	4.98
0.2	9.90
0.4	20.0
0.6	30.0

human torso and head, performing a random motion. Examples of the real video stream, and the reconstruction made with our algorithm is shown in Figure 5.7. On the reconstruction, the torso markers are represented as blue circles, the head markers as red circles, and the joint centre as the large red circle with lines.

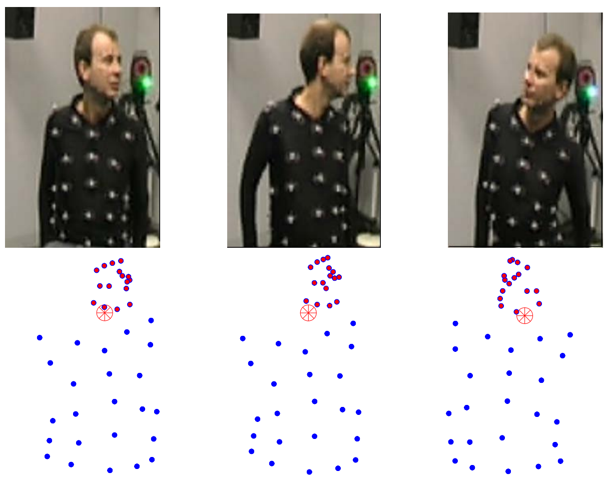


Figure 5.7: On top, sample frames of the real sequence. On the bottom, the corresponding frames of the reconstructed motion. The points of the torso are represented in blue circles, the points of the head as read circles and the joint centre represented as the large red circle with lines.

Even though in the neck there exists a much more complex set of articulations, in this case a universal joint model seems to be a good model. While it may not be detailed enough for clinical applications, in a gross analysis a universal joint model for the neck provides a good example of the applications of these algorithms, as it is clearly capable of determining what we would expect to be the joint centre.

5.3 Hinge joint

Using the same approach as in sections 5.1 and 5.2, we first measure the performance of the algorithm using synthetic data. In this case we simulated a book-like scene, composed of two parallelepipedic objects, with 90 feature points each, linked by a hinge joint. With this setup, we created 1000

completely random motions, with 500 frames each. Each object has 90 feature points. This synthetic hinge joint used is represented in Figure 5.8.

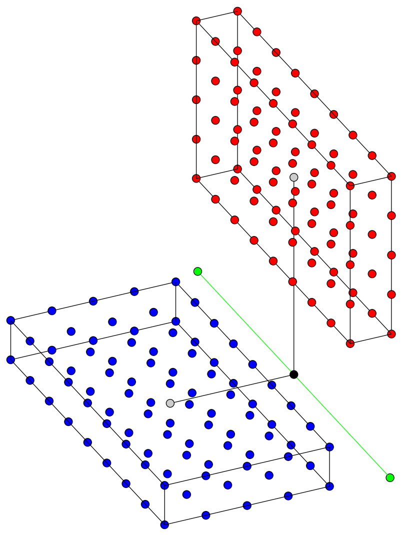


Figure 5.8: A MATLAB plot of the synthetic setup of the hinge joint. The first body's feature points are represented as blue dots. The second body's feature points are represented as red dots; the object centroids are represented as gray dots; the vectors $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$ are represented as black lines; the joint centre is represented as a black dot and the joint axis is represented as a green line. The edges of the objects represented as black lines were added to facilitate visualisation and they are not part of the computations

5.3.1 Performance measurement

To measure the performance of the algorithm we will, once again, rely on the computation of the joint parameters for they depend on all the other parameters of the motion. In this particular case, we want to estimate an axis of rotation, therefore a meaningful measurement for the error is the angle between the ground truth axis and the estimated axis. In Section 2.2.2 we defined the axis of rotation as coincident with the x-axis on the local referential of the first body. Let us define $\mathbf{a}_i^{(gt)}$ as the unitary vector that represents the axis of rotation, in the ground truth data, at frame i. Since in the local referential of the first body the rotation axis is coincident with the x-axis, we can define $\mathbf{a}_i^{(gt)}$ as:

$$\mathbf{a}_i^{(gt)} = \mathtt{R}_i^{(gt1)} [1\,0\,0]^T , \tag{5.14}$$

where $\mathtt{R}_i^{(gt1)}$ is the global rotation matrix in frame i for the first body. We can similarly define $\mathbf{a}_i^{(rec)}$ as the hinge joint axis for the reconstructed case, with $\mathbf{a}_i^{(rec)}$ given by:

$$\mathbf{a}_i^{(rec)} = \mathtt{R}_i^{(r1)} [1\,0\,0]^T , \tag{5.15}$$

where $\mathtt{R}_i^{(r1)}$ is the global rotation matrix in frame i for the first body.

Since we defined $\mathbf{a}_i^{(gt)}$ and $\mathbf{a}_i^{(rec)}$ as unitary vectors, the angle between the vectors at frame i, θ_i, can be given by:

$$\theta_i = \arccos(\mathbf{a}_i^{(gt)} \cdot \mathbf{a}_i^{(rec)}). \tag{5.16}$$

However, this formulation is ambiguous. If the rotation axis was to be defined as anti-parallel to the x-axis, it will still satisfy the same equations. In practice, this is not troublesome as it is just a matter of fixating the referential, and we also do not know the true rotation axis. Still, when we need to measure the angle between the ground truth and reconstructed axis, we must assure the correctness of the axis orientation. Thus, we define $\mathbf{a}_i^{(rec)}$ as the vector, among the two possible vectors, that yields the smaller estimation of θ. Finally, we estimate the error measurement for the hinge joint as the average angle between the ground truth and reconstructed axes over all the F frames:

$$\varepsilon^{(H)} = \frac{1}{F}\sum_{i=1}^{F} |\theta_i| \,. \tag{5.17}$$

5.3.2 Hinge joint factorization with additive Gaussian nose

By using equation 5.17 we can now carry out a statistical analysis of the performance of the algorithm. This is done by running the hinge joint factorization algorithm to the test battery, composed of 1000 completely random hinge joint motions, using 7 different levels of AWGN. Again, the noise levels are defined based on the variance of the Gaussian distribution, with values: $\sigma^2 = 0$, 0.01, 0.05, 0.1, 0.2, 0.4 and 0.6. The statistical results are represented by a box plot on Figure 5.9. Table 5.3.2 presents the median value of $\varepsilon^{(H)}$ in each test, versus the level of noise.

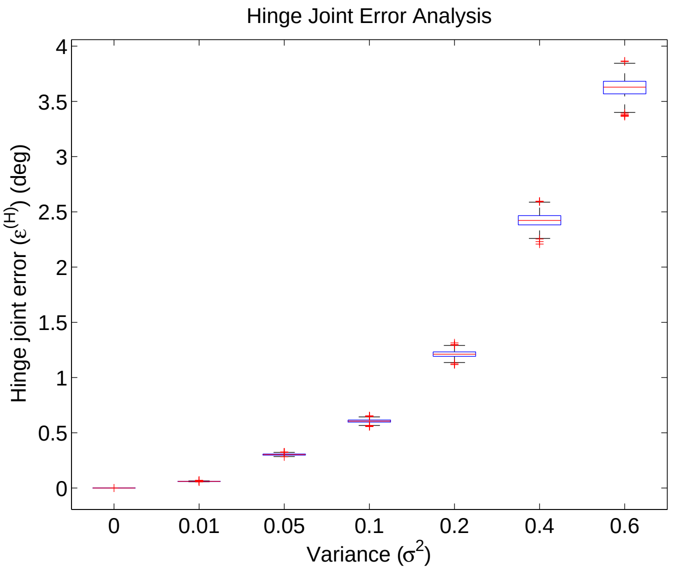


Figure 5.9: A box plot for the Hinge Joint Error angle with 7 different levels of noise.

Notice that the axis of rotation is defined by two parameters: the direction, given by equation (5.15), and the joint centre, given by $\mathbf{d}^{(1)}$. When evaluating the uncertainty on the axis reconstruction, we must both. Still, the uncertainty on the joint centre location was already computed for the universal joint, as the equations that define it are the same in both cases. Therefore we will only analyse the direction of the reconstructed axis.

Figure 5.9 shows that the algorithm presents good convergence, as there is a very small IQR in every test, while the few existent outliers are very close to the data limits. The values for the average error angle between the ground truth and reconstructed axis are presented in Table 5.3. For higher noise levels ($\sigma^2 \geq 0.2$), mean error angle seems to increase linearly with the variance of the AWGN. Still, these error levels are quite high, as the cube was built with side of 2 units.

Table 5.4: Mean error in the Hinge Joint case vs. Noise Level

σ^2	$\varepsilon^{(H)}_{median}$ (degrees)
0	5.18×10^{-6}
0.01	7.1×10^{-3}
0.05	3.2×10^{-1}
0.1	6.1×10^{-1}
0.2	1.2
0.4	2.4
0.6	3.6

5.3.3 Real data

For the illustration of the hinge joint factorization algorithm applied to real data, we will use data captured in the Augmented Human Interaction Laboratory of the Department of Computer Science of Queen Mary, University of London. Again, this data was given by a VICON commercial MOCAP system. The data represents an arm exploring the range of motion of the elbow articulation. In Figure 5.10, we present frames from the real images, compared to the correspondent reconstruction of our algorithm, in roughly the same camera orientation. The hinge joint axis is represented in green, while the two bodies composing the joint are represented in blue and red respectively.

While there is no ground truth data to compare this results, the location of the axis is consistent with what is expected from that type of motion. Given the accuracy of the algorithm with synthetic data, it is expected that, apart from model deviations (*i.e.* human articulations not being completely described by the joint models) the recovered parameters should also have good accuracy.

5.4 Multiple joints

Now that we have exemplified the application of the algorithms for each of the different kinds of joints, we can combine them to analyse a full human body. The data used in this section was obtained from the freely available database of the Graphic Lab of Carnegie-Mellon University. We will apply the

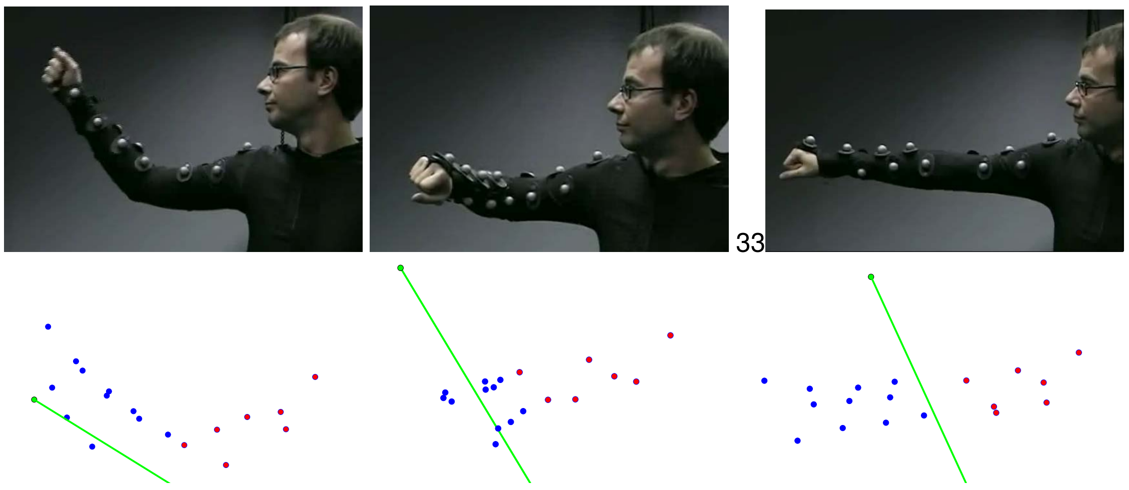


Figure 5.10: On top, some sample frames of the real sequence. On the bottom, the corresponding frames of the reconstructed motion, with the joint axis represented in green.

algorithms to two sequences of very common motions: jogging and kicking a football. In these motions we modelled the knee and elbow articulations as a hinge joint, and the ankle articulation as a universal joint. Other articulations were not modelled because the particular placement of the markers on the subject made hand segmentation difficult.

In Figure 5.11 we present a sequence of frames from the reconstruction of the motion of a human subject jogging. For the hinge joints we represent the rotation axis in green, and for the universal joints we represent the joint centre in red.

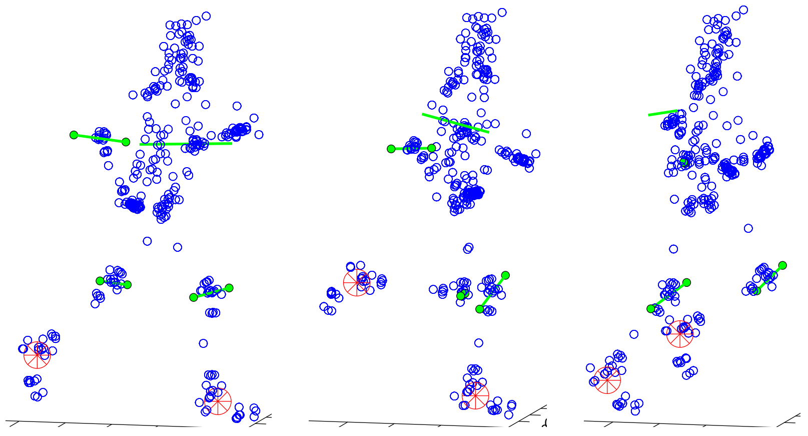


Figure 5.11: Multiple joint parameter estimation during a jogging motion. The Knee and elbow articulations were modelled as hinge joints. Their joint axis are represented in green. The ankle was modelled as a universal joint and its joint centre is represented in red.

In Figure 5.12 we present a sequence of a human subject kicking a football. Again, the joint axes for the hinge joints are represented in green, while the joint centres of the universal joints are represented in red.

As there is no gold-standard method on determining the real axis and joint centres, we can only

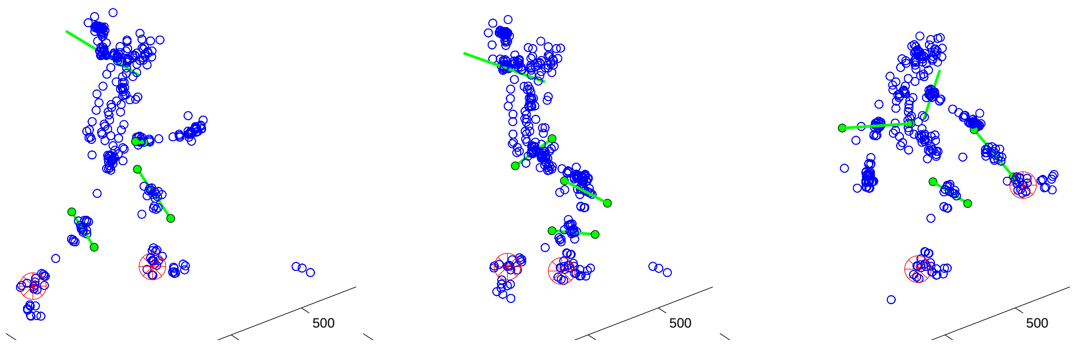


Figure 5.12: Multiple joint parameter estimation of a subject kicking a football. Knee and elbow articulations were modelled as hinge joints. Their axis of rotation is represented in green. The ankle was modelled as a universal joint. The corresponding joint centre is represented in red.

judge them by the graphical representation of the joints. Still, it is clear from Figure 5.11 and Figure 5.12 that the reconstructed joint centres and axes are consistent with what is expected from those articulations.

5.5 Quadratic model

In order to qualitatively evaluate the performance of our quadratic model for non-rigid bodies, we reconstructed a motion of a flexing arm. The model bounds for the deformation values were incorporated on the MATLAB function that is responsible for the non-linear least-squares optimization step on the algorithm, the function *lsqnonlin.m*. As this data was acquired using an elevated number of markers, it provides good information about the soft-tissue deformations. The motion reconstruction was then compared to the reconstruction provided by our weighted factorization approach, in order to evaluate the improvements provided by our model. Images from the reconstruction of the motion using our quadratic model with BA is presented in Figure 5.13, on the top, while the corresponding images of the reconstruction of the motion using the weighted factorization is presented in Figure 5.13, on the bottom.

The analysis on the accuracy of the reconstruction of the motion matrix $\mathtt{W}$ for the forearm shows that the BA minimisation provided a decrease on the error, of about 2 orders of magnitude. For the upper arm, we observed a decrease on the error of 1 order of magnitude. This data is backed up by Figure 5.13, where we can clearly see that the quadratic model and BA provide a more accurate reconstruction of the motion matrix $\mathtt{W}$.

Using the rigid component of the quadratic model, we estimated the parameters of the hinge joint used to model the elbow articulation, and the parameters of the universal joint used to model the shoulder articulation. This data is represented Figure 5.14. As there is no ground truth data for the joint parameters, a qualitative analysis of the figures is the only available method for evaluating the performance of the algorithm. Still, as can be seen on Figure 5.14, both universal and hinge joint

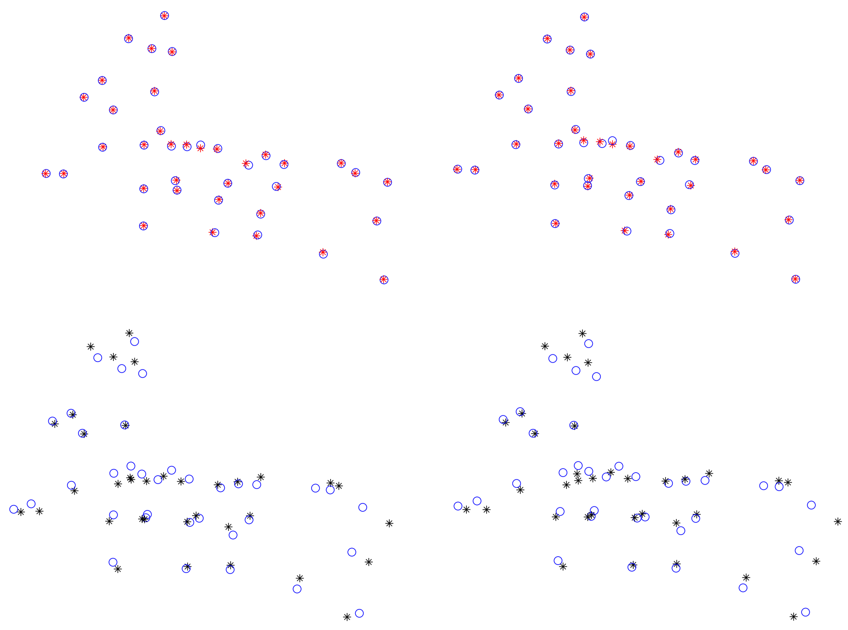


Figure 5.13: Two frames exemplifying the quadratic model with BA, and the weighted factorization approaches on the reconstruction of a non-rigid motion of a human arm. The original data is represented by blue circles on all the images. On the upper images, the quadratic model and BA reconstruction is represented as red asterisks. On the lower images the reconstruction using the weighted factorization is represented by black asterisks.

parameters are consistent with the given motion.

A good motion description and a consistent joint parametrisation are good indicators for the accuracy of this algorithm on modelling articulated non-rigid bodies. Still, a more conclusive analysis can only be done with a proper validation, using ground truth data for the joint parameters. Nonetheless, this algorithm provides promising indicators for its application on this problem.

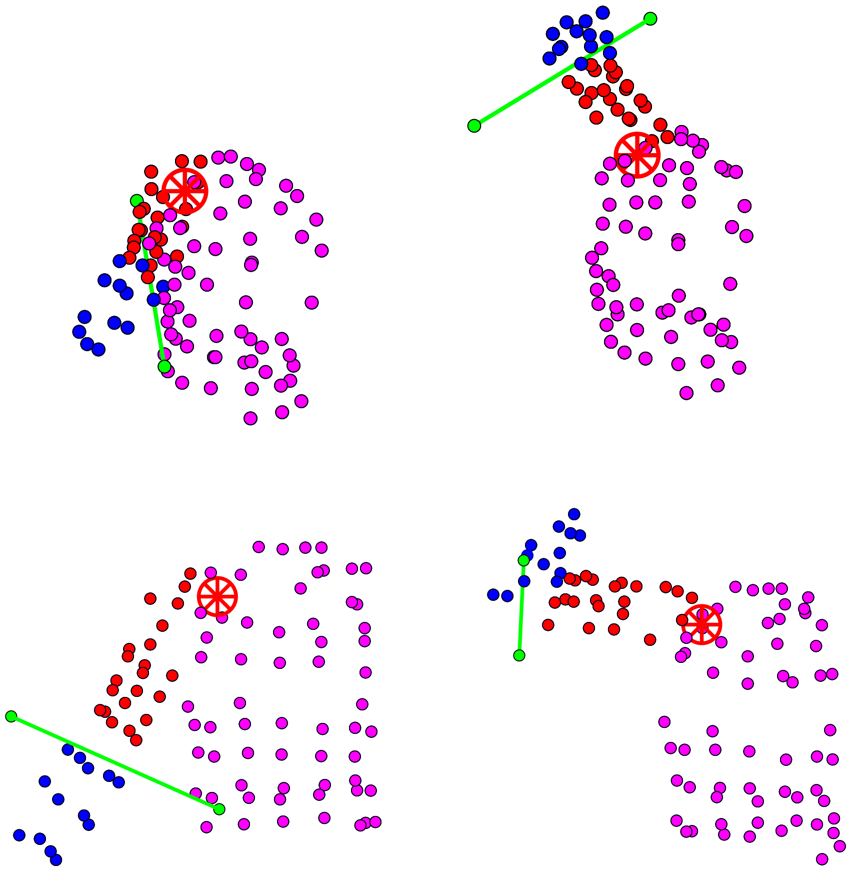


Figure 5.14: Two frames showing the reconstruction of the motion of a human arm. The human torso is represented by magenta dots, while the upper arm is represented by red dots and the forearm is represented by blue dots. The rotation axis of the hinge joint used to model the elbow articulation is represented in green. The joint centre of the universal joint used to model the shoulder articulation is represented by the red circle with lines.

Chapter 6

Conclusion

6.1 Summary

In this thesis we presented algorithms to create articulated 3D human models, based on MOCAP systems, assuming motion segmentation is known. These algorithms rely on an extended formulation of existing SfM algorithms for sets of 2D images to the 3D case.

We then developed a weighted factorization method, which penalises highly deformable points, in order to retrieve a more accurate rigid body description of non-rigid bodies. In the case of non-rigid bodies, this algorithm proved to be more accurate than the existent non-weighted factorization approaches. The weighted factorization algorithm was also extended to deal with occlusion, being shown, by analysis of synthetic data, that it can successfully deal with up to 50% of missing data.

Our method for retrieving the joint parameters of articulated rigid systems, linked by universal and hinge joints, was evaluated using synthetic data with different levels of noise. It proved to return accurate parameters, consistent with the motion observed. Real data was also used for a qualitative analysis on the performance of these algorithms, giving satisfactory results.

To deal with soft-tissue artifacts, we proposed a new quadratic model for non-rigid bodies. A qualitative analysis of this model was done by applying it to real data describing an articulated scene composed of non-rigid bodies. The results show that the joint parameters obtained are consistent with the motion, and the motion reconstruction showed improvements of 1 to 2 orders of magnitude on the error, when compared to the weighted factorization approach.

6.2 Future work

This work provides an insight about the potential applications of structure from motion algorithms in the field of biomechanics. Although not thoroughly validated, our quadratic model for non-rigid bodies gives promising indications towards its applicability on this subject. Naturally, there is room for improvement, and new research paths to follow. We would like to emphasise the following.

Automatic motion segmentation is far from being solved [49]. In this work we assumed segmentation is known. Still, a full automatic method for building human articulated models will need to incorporate automatic segmentation methods.

The new quadratic model for non-rigid bodies introduced in this thesis has given promising indications about its ability to model the non-rigid properties of the human soft-tissue. However, the role of the different types of deformations discussed and their upper and lower bounds need to be further studied in order to lead to a refined model. Additionally, new approaches for parameter estimation, *i.e.,* computational optimization techniques, should also be studied.

Although we tested our algorithms with synthetic data with relative success, on real data we were only able to perform a qualitative analysis. It is of focal interest to validate our approach in a clinical scenario with the appropriate setup. The real data validation is still an open issue which solution could lead to an important breakthrough of this technique for clinical analysis.

Bibliography

[1] Miguel T. Silva. *Human Motion Analysis Using Multibody Dynamics and Optimization Tools*. PhD thesis, Instituto Superior Técnico, July 2003.

[2] M. De Jager. *Mathematical Head-Neck Models for Acceleration Impacts*. PhD thesis, Universiteit Eindhoven, 2000.

[3] R. Huiskes, H. Weinans, and Dalstra M. Adaptive bone remodeling and biomechanical design considerations for noncemented total hip arthroplasty. *Journal of Biomechanics*, 1989.

[4] L. Herda. *Using Biomechanical Constraints to Improve Video Based Motion Capture*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2003.

[5] Eugene J. Alex, Christoph Bregler, and Thomas P. Andriacchi. Tech science press paper galley proof only please return in 48 hours. non-rigid modeling of body segments for improved skeletal motion estimation.

[6] Richard Baker. Gait analysis methods in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 3(1):4, 2006.

[7] Duane Knudson. *Fundamentals of Biomechanics*. Springer Science, second edition edition, 2007.

[8] Artur V. Ferreira, Bruno G. Rosa, J. Fayad, and Miguel T. Silva. Análise de dinâmica directa do movimento de natação: Crawl. In *Conferência Nacional de Dinâmica de Sistemas Multicorpo, Guimarães.*, pages 269–274(6), December 2007. In Portuguese.

[9] Moeslund T.B. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268(38), March 2001.

[10] Sang Il Park and Jessica K. Hodgins. Capturing and animating skin deformation in human motion. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 881–889, New York, NY, USA, 2006. ACM.

[11] Nixon M.S. New advances in automatic gait recognition. *Information Security Technical Report*, 7:23–35(13), December 2002.

[12] Behzad Dariush. Human motion analysis for biomechanics and biomedicine. *Mach. Vision Appl.*, 14(4):202–205, 2003.

[13] A. D. Kuo. A least-squares estimation approach to improving the precision of inverse dynamics computations. *Journal of Biomechanical Engineering*, 120(1):148–159, 1998.

[14] J. R. Gage, P.A. Deluca, and T.S. Renshaw. Gait analysis: Principle and applications with emphasis on its use in cerebral palsy. *J Bone Joint Surg Am.*, (77):1607–1623, 1995.

[15] C.F. Runge, III Zajac, F.E., J.H.J. Allum, D.W. Risher, Jr. Bryson, A.E., and F. Honegger. Estimating net joint torques from kinesiological data using optimal linear system theory. *Biomedical Engineering, IEEE Transactions on*, 42(12):1158–1164, Dec. 1995.

[16] A. Cappozzo, A. Cappello, U.D. Croce, and F. Pensalfini. Surface-marker cluster design criteria for 3-d bone movement reconstruction. *Biomedical Engineering, IEEE Transactions on*, 44(12):1165–1174, Dec. 1997.

[17] Sati M. Quantitative assessment of skin-bone movement at the knee. *The Knee*, 3:121–138(18), August 1996.

[18] A. Leardini, A. Cappozzo, F. Catani, S. Toksvig-Larsen, A. Petitto, V. Sforza, G. Cassanelli, and S. Giannini. Validation of a functional method for the estimation of hip joint centre localtion. *Journal of Biomechanics*, (32):99–103, 1999.

[19] E. J. Alexander and T. P. Andriachhi. Correcting for deformation in skin-based marker systems. *Journal of Biomechanics*, (34):355–361, 2001.

[20] A. Leardini, L. Chiari, U. Della Croce, and Cappozzo A. Human movement analysis using stereophotogrammetry. part 3. soft-tissue artifact assessment and compensation. *Gait and Posture*, (21):212–225, 2005.

[21] T. P. Andriacchi, Alexander E. J., M.K. Toney, C.O. Dyrby, and J. Sum. A point cluster method for *in vivo* motion analysis: Applied to a study of the knee kinematics. *Journal of Biomechanical Engineering*, 120(12):743–749, 1998.

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[23] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *Lecture Notes in Computer Science*, 800:97–110, 1994.

[24] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159 – 179, 1998.

[25] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina*, pages 690–696, June 2000.

[26] M. Brand and R. Bhotika. Flexible flow for 3d nonrigid tracking and shape recovery. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii*, pages 315–22, December 2001.

[27] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proc. 8th European Conference on Computer Vision, Copenhaguen, Denmark*, pages 573–587, May 2004.

[28] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet–based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, March 2007.

[29] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 1:677–684 vol.1, 2000.

[30] C. Sminchisescu and B. Triggs. Covariance scaled smapling for monocular 3d body tracking. In *Proc. CVPR, vol. I*, pages 447–454, 2001.

[31] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Diego, California*, volume 2, pages 1110–1115, June 2005.

[32] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Diego, California*, volume 2, pages 815–821, June 2005.

[33] P. J. Rbmann and F. T. Sheehan. Precise 3d skeletal kinematics using fast phase contrast magnetic resonance imaging. *Journal of Magnetic Resonance Imaging*, (17):206–213, 2003.

[34] D. S. Asakawa, G. P. Pappas, S. S. Blemker, J. E. Drace, and Delp S. L. Cine phase contrast magnetic resonance imaging as a tool for quantification of skeletal muscle motion. *Seminars on Muskuloskeletal Radiology*, (7):287–295, 2003.

[35] J. Fayad, A. Del Bue, and P.M.Q. Aguiar. Articulated motion analysis from motion capture data. In *8th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering (CMBBE 2008)*, February 2008.

[36] J. Yan and M. Pollefeys. Articulated motion segmentation using ransac with priors. *ICCV Workshop on Dynamical Vision*, 2005.

[37] Pedro M.Q. Aguiar and José M.F. Moura. Factorization as a rank 1 problem. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1178, 1999.

[38] P.M.Q. Aguiar and J.M.F. Moura. Weighted factorization. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 1:549–552 vol.1, 2000.

[39] M. Irani and P. Anandan. Factorization with uncertainty. In *Proc. 6th European Conference on Computer Vision, Dublin, Ireland*, pages 539–553, 2000.

[40] Pedro M.Q. Aguiar and José M.F. Moura. Rank 1 weighted factorization for 3d structure recovery: Algorithms and performance analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1134–1049, 2003.

[41] Kenichi Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science Inc., New York, NY, USA, 1996.

[42] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, 2007.

[43] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, 2005.

[44] Y. C. Fung. *Biomechanics: Mechanical Properties of Living Tissue*. Springer-Verlag, 1993.

[45] K. B. Atkinson, editor. *Close Range Photogrammetry and Machine Vision*. Engineering and Science. Whittles Publishing, 1996.

[46] B. Triggs, P. McLauchlan, R. I. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[47] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[48] B. Triggs. Auto-calibration and the absolute quadric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 609–614, 1997.

[49] K. Kanatani. Motion segmentation by subspace separation and model selection. In *Proceedings of the 8th International Conference on Computer Vision*, volume 2, pages 301–306, Vancouver, Canada, July 2001.

Appendix A

Mathematical Formulation for the Weighted Factorization Algorithm

Let us consider the least-squares problem of finding a general $m \times k$ matrix $\mathtt{X}$ such that:

$$\arg\min_{\mathtt{X}} \; ||\mathtt{A}\,\mathtt{X} - \mathtt{B}|| \,. \tag{A.1}$$

where $\mathtt{A}$ is a general $n \times m$ full-rank matrix, and $\mathtt{B}$ is a general $n \times k$ matrix. It can be shown that the solution to the least-squares problem is given by equation (A.2)[ref]:

$$\begin{aligned} \mathtt{A}\,\mathtt{X} &= \mathtt{B} \\ (\mathtt{A}^T\mathtt{A})\,\mathtt{X} &= \mathtt{A}^T\,\mathtt{B} \\ \mathtt{X} &= (\mathtt{A}^T\,\mathtt{A})^{-1}\,\mathtt{A}^T\,\mathtt{B}. \end{aligned} \tag{A.2}$$

Let us now return to the least squares problem defined by equation (2.34), in the case were we assume to known $\mathtt{M}$. Since the weight matrix $\mathtt{C}_j$ is the inverse co-variance matrix of the 3D coordinates of point j, it has full-rank and it is positive definite. Thus we can use Cholesky decomposition to write:

$$\mathtt{C}_j = \mathtt{P}_j\,\mathtt{P}_j^T \,. \tag{A.3}$$

Using this result together with equation (2.34) we now write:

$$\begin{aligned}
&\arg\min_{\mathtt{S}} \sum_{i=1}^{i=F}\sum_{j=1}^{j=P}(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{M}_i\mathbf{s}_j)^T\,\mathtt{P}_j\mathtt{P}_j^T\,(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{M}_i\mathbf{s}_j) \Leftrightarrow \\
\Leftrightarrow\quad &\arg\min_{\mathtt{S}} \sum_{i=1}^{i=F}\sum_{j=1}^{j=P}[\mathtt{P}_j^T(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{M}_i\mathbf{s}_j)]^T\,[\mathtt{P}_j^T\,(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{M}_i\mathbf{s}_j)] \Leftrightarrow \\
\Leftrightarrow\quad &\arg\min_{\mathtt{S}}) \sum_{i=1}^{i=F}\sum_{j=1}^{j=P}(\mathtt{P}_j^T\,\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{P}_j^T\,\mathtt{M}_i\mathbf{s}_j)^T\,(\mathtt{P}_j^T\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{P}_j^T\mathtt{M}_i\mathbf{s}_j) \Leftrightarrow \\
\Leftrightarrow\quad &\arg\min_{\mathbf{x}} \left|\left|\mathtt{B}^{(S)} - \mathtt{A}^{(S)}\,\mathbf{x}^{(S)}\right|\right|, \qquad \text{(A.4)}
\end{aligned}$$

where $\mathbf{b}_{ij}^{(S)} = \mathtt{P}_j^T\tilde{\mathbf{w}}_{ij}$, $\mathtt{A}_{ij}^{(S)} = \mathtt{P}_j^T\,\mathtt{M}_i$ and $\mathbf{x}_j^{(S)} = \mathbf{s}_j$. Note that $||\mathtt{B} - \mathtt{A}\,\mathbf{x}|| = ||\mathtt{A}\,\mathbf{x} - \mathtt{B}||$. Thus we can now apply the result derived in equation (A.2). Writing the correspondent equations for each vector $\mathbf{x}_j$, and substituting for $\mathbf{s}_j$, $\tilde{\mathbf{w}}_{ij}$, $\mathtt{C}_j = \mathtt{P}_j\,\mathtt{P}_j^T$ and $\mathtt{M}_i$ we will get equation (2.36). Based on analogous steps, equation (2.39) can also be proved:

$$\begin{aligned}
&\arg\min_{\mathtt{m}_i} \sum_{i=1}^{i=F}\sum_{j=1}^{j=P}(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{S}_j\mathbf{m}_i)^T\,\mathtt{P}_j\mathtt{P}_j^T\,(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{S}_j\mathbf{m}_i) \Leftrightarrow \\
\Leftrightarrow\quad &\arg\min_{\mathtt{m}_i} \sum_{i=1}^{i=F}\sum_{j=1}^{j=P}[\mathtt{P}_j^T(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{S}_j\mathbf{m}_i)]^T\,[\mathtt{P}_j^T\,(\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{S}_j\mathbf{m}_i)] \Leftrightarrow \\
\Leftrightarrow\quad &\arg\min_{\mathtt{S}}) \sum_{i=1}^{i=F}\sum_{j=1}^{j=P}(\mathtt{P}_j^T\,\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{P}_j^T\,\mathtt{S}_j\mathbf{m}_i)^T\,(\mathtt{P}_j^T\tilde{\mathbf{w}}_{ij}^{(r)} - \mathtt{P}_j^T\mathtt{S}_j\mathbf{m}_i) \Leftrightarrow \\
\Leftrightarrow\quad &\arg\min_{\mathbf{x}} \left|\left|\mathtt{B}^{(M)} - \mathtt{A}^{(M)}\,\mathbf{x}^{(M)}\right|\right|, \qquad \text{(A.5)}
\end{aligned}$$

where $\mathbf{b}_{ij}^{(M)} = \mathtt{P}_j^T\tilde{\mathbf{w}}_{ij}$, $\mathtt{A}_{ij}^{(M)} = \mathtt{P}_j^T\,\mathtt{S}_j$ and $\mathbf{x}_j^{(M)} = \mathbf{m}_i$. Applying the result derived on A.2 and substituting for $\mathbf{m}_i$, $\tilde{\mathbf{w}}_{ij}$, $\mathtt{C}_j = \mathtt{P}_j\,\mathtt{P}_j^T$ and $\mathtt{S}_j$, we will get equation (2.36).

Appendix B

Procrustes Analysis

Procrustes Analysis is method for registering two shapes, represented by point clouds, using only a rotation and a scale factor. In our case we will only deal with the rotation factor, as scale is treated in another way in the algorithm. Consider a general $3 \times k$ matrix $\mathtt{B}$ and a general $3 \times k$ matrix $\mathtt{A}$. We want to find the best 3×3 rotation matrix $\mathtt{R}$ that satisfies the condition:

$$\arg\min_{\mathtt{R}} \, ||\mathtt{A} - \mathtt{R}\,\mathtt{B}|| \, . \tag{B.1}$$

Since we are using the Frobenius norm, we know that $||\mathtt{X}|| = trace(\mathtt{X}^T\mathtt{X})$. Applying this result to equation (B.1), and using the properties of the $trace$ operator we get:

$$||\mathtt{A} - \mathtt{R}\,\mathtt{B}|| = trace(\mathtt{A}^T\mathtt{A} + \mathtt{B}^T\mathtt{B}) - 2 \times trace(\mathtt{B}\,\mathtt{A}^T\,\mathtt{R}). \tag{B.2}$$

Since only the second term depends on $\mathtt{R}$, the registration problem can also be stated as:

$$\arg\max_{\mathtt{R}} trace(\mathtt{B}\,\mathtt{A}^T\,\mathtt{R}). \tag{B.3}$$

Let us use the SVD to define $\mathtt{B}\,\mathtt{A}^T = \mathtt{U}\,\mathtt{S}\,\mathtt{V}^T$. From the cyclic property of the $trace$ operator we get:

$$\begin{aligned} trace(\mathtt{B}\,\mathtt{A}^T\,\mathtt{R}) &= trace(\mathtt{U}\,\mathtt{S}\,\mathtt{V}^T\,\mathtt{R}) && \text{(B.4)} \\ &= trace(\mathtt{S}\,\mathtt{V}^T\,\mathtt{R}\,\mathtt{U}) && \text{(B.5)} \\ &= trace(\mathtt{S}\,\mathtt{H}). && \text{(B.6)} \end{aligned}$$

Note that $\mathtt{R}$, $\mathtt{U}$ and $\mathtt{V}$ are orthogonal matrices, and so $\mathtt{H}$ is also an orthogonal matrix. By the definition of trace we have that:

$$trace(\mathtt{S}\,\mathtt{H}) = \sum_{i=1}^{3} s_i \times h_{ii}. \tag{B.7}$$

Note that the singular values s_i are all non-negative. Being $\mathtt{H}$ an orthogonal matrix, $trace(\mathtt{S}\,\mathtt{H})$ is maximum when $h_{ii} = 1$ *i.e.* when $\mathtt{H}$ is the 3×3 identity matrix. Thus we can finally define $\mathtt{R}$ as:

$$\mathtt{R} = \mathtt{U}\,\mathtt{V}^T. \tag{B.8}$$