Universitat Autònoma de Barcelona

Missing Data Matrix Factorization Addressing the Structure from Motion Problem

A dissertation submitted by **Ma Carme Julià Ferré** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, February 2008

Director: **Dr. Angel D. Sappa**
Computer Vision Center
Co-director: **Dr. Felipe Lumbreras**
Universitat Autònoma de Barcelona
Dept. Ciències de la Computació & Computer Vision Center

Centre de Visió
per Computador

This document was typeset by the author using $\LaTeX\,2_{\varepsilon}$.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN 84–922529–8–7

Printed by Ediciones Gráficas Rey, S.L.

als meus pares

Acknowledgements

Primer de tot, voldria donar les gràcies als meus directors: l'Angel Sappa i en Felipe Lumbreras (o en Felipe i l'Angel, cap dels dos vol anar primer). Gràcies per la vostra disponibilitat en tot moment i els ànims continus. A l'Angel li vull agrair els ànims i lo bé que sap transmetre optimisme i sobretot, autoestima. La seva paciència i el seu positivisme donen energia per tirar endavant. L'anglès seria molt diferent sense la teva ajuda, sempre ajudes a fer convergir. A Felipe, li voldria agrair que sempre doni idees i experiments per provar. I que m'hagi ensenyat a ordenar-los. I els mails des de l'espai cibernètic sempre animaven molt. Ha estat molt i molt agradable treballar amb vosaltres. Moltes gràcies, de veritat. Aquest treball no hauria estat possible sense la vostra ajuda, disponibilitat, idees i ànims.

Voldria agrair al grup ADAS l'haver sobreviscut en més d'un assaig de presentació i haver-me donat consells per millorar. Gràcies Dani, pels dubtes burocràtics, els plots tan guais i els consells múltiples. Jose, gràcies per l'interés i pels consells. Ah, i per fer respectar el silenci de la sala algun cop puntual.

A la gent del CVC en general i sobretot al grup amb els qui m'he fet més: Enric i Jaume (gràcies per comprendre el meu tancament hermètic i per haver cregut sempre en el Julià-sytem), a tota la gent de la sala, gràcies pel bon ambient de treball: Àgata (ha estat guai compartir el cubicle), Agnés (sempre amb els palitos a punt), Xevi (tan didàctic amb els disbarats que pregunto) i David. A la gent que heu anat passant pel CVC: Oriol (per escoltar-me explicant rareses matemàtiques, animar-me i aconsellar-me), Dani Rowe i Poal (per l'ajuda amb el papeleo!), Batlle (amb les seves teories, sempre divertides), Anton. I algú més que segur que em deixo (sense voler).

Voldria felicitar la Raquel i el Joan per no morir d'un atac en preguntar-los coses rares i presentar-los nous virus i fenòmens paranormals que només em passen a mi. També voldria agrair a Xavi Roca per la paciència amb les firmes, els ànims i l'apoio sempre. A Ramon li voldria agrair els consells, les xerrades, la piscina (ja n'hi me'n recordo de quan venies) i la preocupació en els moments delicats.

No em puc deixar en Francesc, perquè tot i l'aparença, tots sabem que és un corderet i que està sempre allí per donar un cop de mà. Gràcies per l'arròs al forn i el mullauret (que no sé si s'escriu així) i pels ànims, la música i els consells en tot moment.

Especialment, voldria donar les gràcies a l'Anna Salvatella pels ànims i per la capacitat que té de trobar solucions i camins per als problemes. De vegades han estat del tot determinants. I fora del CVC, voldria agrair a l'Anna i en Marc que m'hagin donat l'oportunitat de conéixer gent tan especial com els joves, que tants moments bonics saben regalar.

Ani, gràcies per estar sempre tan a prop i per transmetre tan optimisme fins i tot via mail. A l'Anna Turu li voldria agrair les xerrades durant els dinars a la UAB. S'han trobat molt a faltar. Tenim pendent més d'un pingado, ho sé.

I also would like to thank to Dr. René Vidal, Dr. Pedro Carmona, Aeron Buchanan and Ira Kemelmacher, for providing me with datasets, code and interesting insights.

He de recordar a Alessandro Baricco i a Jesús Moncada, per ajudar a fer més curts els trajectes (sempre ràpids i puntuals) de la renfe.

A la meua família, els meus fans incondicionals: les meues cosines (Ma Mar i Pili) i cosins polítics (Xavi i Xato), al meu cosí preferit, Cisco (sempre a punt per vindre a sopar), als meus padrins (Francisco i Francisca) i als meus iaos, gràcies per creure sempre en mi. També agrair a la meua segona família (”los carteros”), els ànims en este sprint final, sobretot a l'Eugenio, que sempre ha cregut en mi. No em puc deixar a la Josefina, sempre mostrant interés i guardant coses bones del poble per donar-me el cap de setmana.

A la meua germana Tresa, pels ànims, les xarrades nocturnes (com se troben a faltar!), los ratets d'estudi al cosidó, les comèdies i els mails. I al meu cunyat Roca, per interessar-se per les coses rares de la pantalla i per aguantar algun assaig de presentació.

Als meus pares, Miquel Àngel i Tresa, agrair-los l'estar sempre al meu costat, els ànims i l'aguantar-me tot este temps, des de que vaig marxar de casa, que no és poc! Gràcies per permetre'm el luxe d'estudiar tot este temps, mimant-me cada cap de setmana i omplint fiambreres amb los millors suquets. Este treball no haurie segut possible sense el vostre apoio constant. Al final, pare, hai ficat ”the end”, eh?

Finalment, voldria donar les gràcies a Joan Carles (que hauria d'aparéixer com a segon autor), per moltes coses. Sobretot, per la paciència que només ell té, per aguantar-me, per cuidar-me, per animar-me contínuament, per haver vingut a la gran capital i per fer-me tan feliç.

Moltes gràcies a tots.

Aquest treball ha estat finançat pel Govern Espanyol, amb els projectes CICYT TRA2004-06702/AUT, la MEC TRA2007-62526/AUT i el programa Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

Abstract

This work is focused on the missing data matrix factorization addressing the *Structure from Motion* (SFM) problem. The aim is to decompose a matrix of feature point trajectories into the *motion* and *shape* matrices, which contain the relative camera-object motion and the 3D positions of the tracked feature points, respectively. This decomposition can be found by using the fact that the matrix of trajectories has a reduced rank. Actually, when the trajectories belong to a single object, the rank of the matrix is at most 4. Although several techniques have been proposed to tackle this problem, they may give undesirable results when the percentage of missing data is very high. An iterative multiresolution scheme is presented to deal with matrices with high percentages of missing data. The idea of the proposed scheme is to consider submatrices with a low percentage of missing data. Then, missing data in these submatrices are filled in, by using a factorization technique. The final goal is to improve the results applying a factorization technique to the filled in matrix with the proposed scheme, instead of applying it directly to the given input matrix, which has a high percentage of missing data. Experimental results show the viability of the proposed approach.

In the multiple objects case, factorization techniques can not be directly applied to obtain the SFM of each of the objects, since the trajectories are not sorted into different objects. Furthermore, another problem should be faced out: the estimation of the rank of the matrix of trajectories. The problem is that, in this case, the rank of the matrix of trajectories is not bounded, since any prior knowledge about the number of objects nor about their motion is used. This problem becomes more difficult with missing data, since the singular values can not be computed to estimate the rank. A technique to estimate the rank of a missing data matrix of trajectories is presented. The intuition behind the proposed technique is that, since feature point trajectories belong to surfaces of rigid objects, the *frequency spectra* of the input matrix should be preserved after recovering missing entries. Missing data are filled in by applying a factorization technique considering different rank values. The rank of the matrix of trajectories is estimated by using a measure of goodness that compares the frequency spectra of each filled in matrix with the one of the input matrix. The good performance of the proposed technique is empirically shown considering sequences with both, synthetic and real data. Once the rank is estimated and the matrix of trajectories is full, the motion segmentation of the trajectories is computed. Finally, any factorization technique for the single object case gives the shape and motion of every object.

In addition to the SFM problem, this thesis also shows other applications that can be addressed by means of factorization techniques. Several problems in which data can be stored into a matrix and that can present missing data are studied, searching for similarities and differences with the SFM problem. The Alternation technique, which is used through the thesis, is adapted to address each particular problem. The first proposed application is the photometric stereo: the goal is to recover the reflectance and surface normals and the light source direction at each frame, from a set of images taken under different lighting conditions. Pixels in shadow and specularities are considered as missing data. Hence, these points do not affect the final results. In a second application, the aim is to fill in missing data in gene expression matrices by using the Alternation technique. These matrices are generated taking information provided by *DNA microarrays*. However, missing data can occur during the data acquisition process. Finally, the Alternation technique is adapted to be applied in recommender systems, widely considered in *E-commerce*. The idea of recommender systems is to predict the rates that a particular customer would give to items, by using information stored in the system. This information consists of rates provided by the current customer and also by the rest of customers that have used the system. For each application, experimental results are given in order to show the good performance of the proposed Alternation-based strategy.

Resum

Aquest treball es centra en la factorització de matrius per obtenir l'Estructura a partir del Moviment (SFM). La idea és descomposar la matriu de trajectòries en la matriu de moviment, que conté la posició relativa càmera-objecte a cada frame, i la matriu de forma, que conté les coordenades 3D dels punts característics. Aquesta factorització es pot obtenir utilitzant que la matriu de trajectòries té un rang reduït. En particular, si les trajectòries pertanyen a un únic objecte rígid, la matriu té com a molt rang 4. Tot i que s'han proposat diverses tècniques per tractar el problema de les matrius amb forats, aquestes poden no donar resultats correctes quan el percentatge de forats a la matriu és molt gran. Proposem un esquema multiresolució iteratiu per poder tractar aquests casos amb molts elements buits a la matriu de trajectòries. L'esquema proposat consisteix en considerar submatrius amb un percentatge de forats reduït. Seguidament, els forats d'aquestes matrius són emplenats aplicant un mètode de factorització. L'objectiu final és obtenir millors resultats aplicant un mètode de factorització a la matriu emplenada amb l'esquema que proposem, enlloc d'aplicar-lo a la matriu inicial, que presenta un elevat percentatge de forats. Els resultats experimentals certifiquen la validesa de l'esquema que proposem.

En el cas de matrius de trajectòries corresponents a punts característics que pertanyen a diversos objectes, les tècniques de factorització no es poden aplicar directament per obtenir el moviment i la forma de cada objecte, ja que les trajectòries no estan ordenades per objectes. A més a més, s'ha de tenir en compte un altre problema: l'estimació del rang de la matriu de trajectòries. El problema és que amb forats, el rang de la matriu no pot ser calculat directament. A més a més, com que hi ha múltiples objectes, és difícil d'estimar-lo, sense fer ús d'informació com ara nombre d'objectes o del tipus de moviment d'aquests. Presentem una tècnica per estimar el rang d'una matriu de trajectòries amb forats. La idea és que, si les trajectòries pertanyen a objectes rígids, la *freqüència espectral* de la matriu de trajectòries inicial no hauria de variar un cop la matriu ha estat emplenada. Els forats de la matriu són emplenats amb un mètode de factorització, considerant diferents valors per al rang de la matriu. Al mateix temps, el rang de la matriu de trajectòries és estimat fent servir una mesura que compara la freqüència espectral de cada matriu emplenada amb la de la matriu inicial. La tècnica d'estimació del rang proposada és validada considerant seqüències sintètiques i reals. El proper pas consisteix en segmentar les trajectòries segons el seu moviment. Finalment, qualsevol tècnica d'Estructura a partir de Moviment (SFM) per a un únic objecte pot ser aplicada per trobar el moviment i la forma de cadascun dels objectes.

Intentem aplicar la metodologia proposada per al problema de l'Estructura a partir de Moviment (SFM) a d'altres aplicacions, no només dins el camp de la visió per computador. En particular, l'objectiu és adaptar els mètodes *Alternats*, usats al llarg de tota la tesi, per poder aplicar-los a diferents problemes. Per a poder dur-ho a terme, diferents problemes són estudiats, buscant les diferències i semblances amb el problema de l'Estructura a partir de Moviment. A continuació, els mètodes Alternats són adaptats de manera que puguin ser aplicats per poder resoldre el problema en concret. Una de les possibles aplicacions és la fotometria: la idea és recuperar la reflectància i les normals a la superfície i la direcció de la llum en cada imatge, a partir d'imatges obtingudes sota diferents condicions de llum. Proposem considerar els píxels que pertanyen a una zona amb ombres o a una zona saturada com a forats a la matriu de dades. D'aquesta manera, aquests punts no afectaran el resultat final. En una segona aplicació, l'objectiu és adaptar els mètodes Alternats per poder omplir els forats en una matriu de dades provinents d'expressions de gens. Aquestes matrius són generades amb la informació que proporcionen els *DNA microarrays*. Aquesta tècnica permet visualitzar tots els gens a la vegada. Finalment, els mètodes Alternats són adaptats i aplicats a matrius de dades de sistemes de recomanació, molt usats en *E-commerce*. Aquestes matrius contenen puntuacions que els usuaris han donat a certs productes. La idea és predir les puntuacions que un usuari concret donaria a altres productes, utilitzant la informació emmagatzemada en el sistema. Aquesta informació ha estat proporcionada pel propi usuari i per tots els que han entrat algun cop al sistema. En totes tres aplicacions, es presenten resultats experimentals per validar les respetives adaptacions dels mètodes Alternats.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Several problems can be reduced to find the closest low-rank matrix approximation of a large matrix. Some of these problems can be found in fields so different such as: computer vision (e.g., *Structure from Motion* [110, 109], optical flow estimation [65]), bioinformatics (e.g., data mining [116, 5]), photometry (e.g., [9, 133]), marketing (e.g., recommender systems [99, 15]), image compression (e.g., [127]) or PCA (e.g., [27]). In fact, in the last case, the aim is to find the closest low-dimension subspace, instead of the closest low-rank matrix.

This thesis is focused on the *Structure from Motion* (SFM) problem, which consists in extracting the 3D object shape as well as the relative camera-object motion from 2D trajectories of tracked features. This problem can be applied in robotics, virtual reality or generation of 3D digital models from real objects.

In computer vision context, the **factorization technique**, which was originally introduced by Tomasi and Kanade [110], is a theoretically sound method addressing the SFM problem. The idea of the factorization technique is to project a set of 3D feature points into the image plane and to track the obtained 2D image coordinates along various frames. These 2D image coordinates are stacked into a *matrix of trajectories* and the aim is to factorize this matrix into the relative camera-object motion and the object shape (3D position of the tracked feature points). That is, factorization aims at expressing the matrix of trajectories (denoted as W) as the product of two unknown matrices, namely, the relative camera-object motion at each frame (M) and the 3D object shape (S): $W = MS$. These matrices are referred to as *motion* and *shape* matrices, respectively.

In the original approach [110], a rigid object is observed by an affine camera. That camera model is also used through this thesis. The key point is that the rigidity of the scene imposes that the matrix of trajectories lies in a low-dimensional subspace. Therefore, the Structure from Motion problem can be reduced to find the closest low-rank approximation to the matrix of trajectories. For instance, in the case of a single rigid moving object, and under affine camera model, the rank of the matrix of trajectories is at most 4. The Singular Value Decomposition (SVD) [43] is, in general, used to obtain the best approximation in the least-squares sense when the matrix of trajectories is full. That is, when the 2D tracked feature points are visible in all the frames.

Fig. 1.1 presents an illustration of factorization applied to the SFM problem. Concretely, Fig. 1.1 (a) shows several frames of a sequence considered for the example. The camera is fixed, while the object rotates around a vertical axis. Feature point trajectories plotted into the image plane are shown in Fig. 1.1 (b). Each feature point trajectory is stacked into each column of the matrix of trajectories W (see Fig. 1.1 (c)). Hence, each row of the matrix correspond to each frame of the sequence. The elements of the matrix of trajectories are represented by means of a grey level scale. The x and y coordinates are stacked separately: the upper part corresponds to the x–coordinates, while the bottom part correspond to the y–coordinates. If the factorization technique is applied to W, the decomposition given in Fig. 1.1 (d) is obtained. Notice that $\widetilde{W}$ is the best rank-4 approximation of W. An example of the factors obtained with the factorization is provided: the relative camera-object motion in the five studied frames (Fig. 1.1 (e)) and the 3D positions of the tracked feature points (Fig. 1.1 (f)).

Several variants of the original factorization approach have been proposed in the literature: some of them are based on more accurate camera models such as scaled orthographic projection [20], weak perspective (e.g., [124, 134]), para-perspective [91] and the purely perspective cameras (e.g., [113, 83]) (details on the different camera models can be found in [55]); others consider non-rigid (e.g., [30, 29]) or articulated objects (e.g, [125, 112]).

1.1 Motivation

In real situations, feature point trajectories can be incomplete or split up due to objects occlusions, missing on the tracking or simply because some features exit the camera field of view. It results in missing entries in the matrix of trajectories. Since the SVD can not be directly used with missing data, other techniques should be considered. Although several approaches have been proposed to tackle the Structure from Motion problem through factorization of a missing data matrix, the problem has not a closed form solution yet. Actually, some of the proposed techniques can not deal with high percentages of missing data.

Fig. 1.2 shows an example of a sequence where some feature points are occluded, giving missing data in the matrix of trajectories. Five frames of the considered sequence are depicted in Fig. 1.2 (a). The camera is fixed, while the dinosaur rotates around a vertical axis. Feature point trajectories plotted into the image plane are shown in Fig. 1.2 (b). The obtained matrix of trajectories, generated as in the previous example, is shown in Fig. 1.2 (c). White points correspond to unknown entries in W. In this example, unknown entries are due to points that are self-occluded by the object, due to the rotation. If the factorization technique is applied to W, the decomposition given in Fig. 1.2 (d) is obtained. Fig. 1.2 (e) depicts the obtained

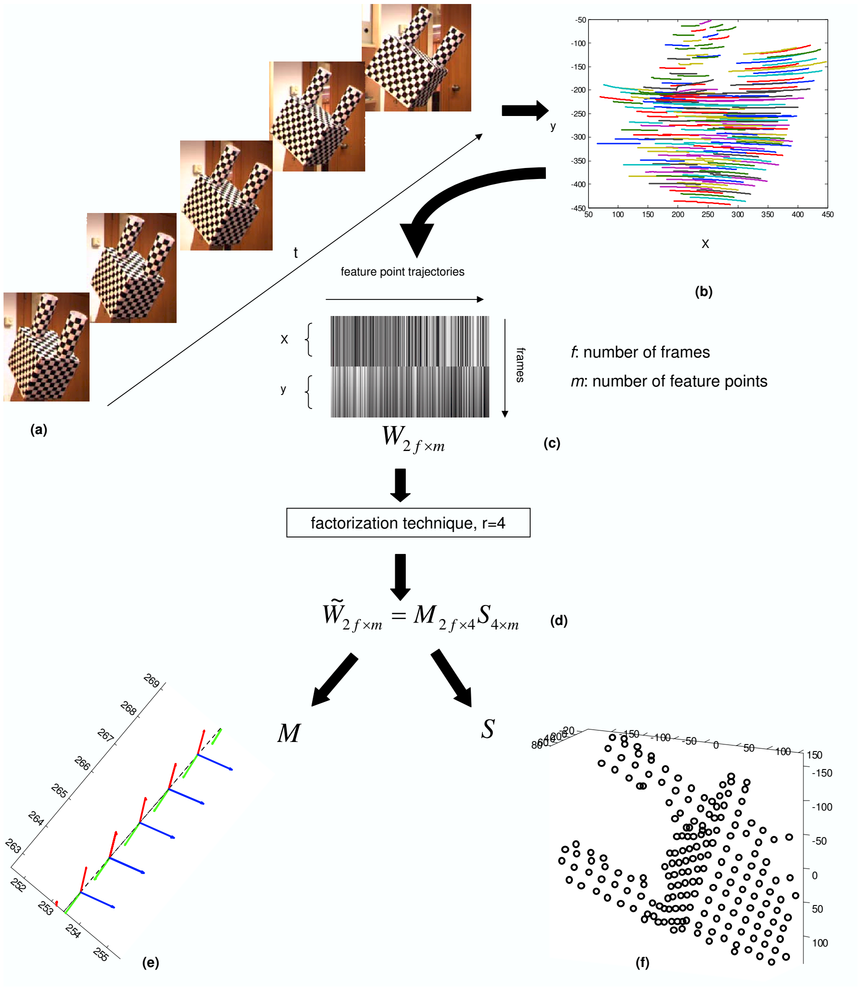


Figure 1.1: (a) five frames of a video sequence; (b) feature point trajectories plotted into the image plane; (c) matrix of trajectories W; (d) obtained decomposition by applying factorization; (e) relative camera object motion, M; (f) 3D positions of the feature points, S.

full trajectories plotted into the image. An example of the factors obtained with the factorization is provided: the relative camera-object motion (Fig. 1.2 (f)) and the 3D positions of the tracked feature points (Fig. 1.2 (g)).

In the multiple objects case, further considerations should be taken into account. First of all, factorization techniques can not be directly used to decompose the matrix of trajectories into the shape and motion matrices; on the one hand, each object has different motion and on the other hand, trajectories are not sorted into different objects in the matrix of trajectories. Therefore, Structure from Motion problem should first segment the matrix of trajectories into different objects, in order to be able to apply a single object technique to obtain the SFM of every object. Secondly, an additional problem should be faced out: the estimation of the rank of the matrix of trajectories. In this case, the rank of the matrix W is not bounded as in the single object case (where the rank is 4 at most), since any prior knowledge of the scene, such as number of objects or kind their motion is used. This problem becomes more difficult when the matrix of trajectories has missing data, since the singular values can not be used to estimate the rank and it can not be directly computed.

1.2 Objective

The main objective of this thesis is to obtain the 3D shape and the relative camera-object motion from a given matrix of trajectories that contains missing data. The factorization technique is used to address this problem. The trajectories can belong to a single or to multiple objects. Furthermore, this thesis also aims at adapting the factorization technique, in particular the *Alternation*, to tackle other problems, apart from SFM.

In summary, this thesis aims at:

- obtaining the SFM by factorizing trajectory matrices that have a high percentage of missing data. To do that, an iterative multiresolution scheme is proposed, focused on the single object case. The underlying idea of the proposed approach is to consider submatrices with a low percentage of missing data. Then, a factorization technique can be applied to fill in missing data in these submatrices. The goal is to improve results when a factorization technique is applied to the filled in matrix with the proposed approach, instead of to the original matrix, which has a high percentage of missing data.

- recovering the shape and motion from a missing data matrix of trajectories, which belong to different rigid objects. First of all, a novel technique to estimate the rank of the missing data matrix of trajectories is presented. Then, the trajectories are segmented into different objects. Finally, the shape and motion of every object contained in the scene can be recovered, by applying any single object technique.

- applying factorization techniques to tackle problems apart from SFM. Concretely, adaptation of the Alternation technique to address problems in the

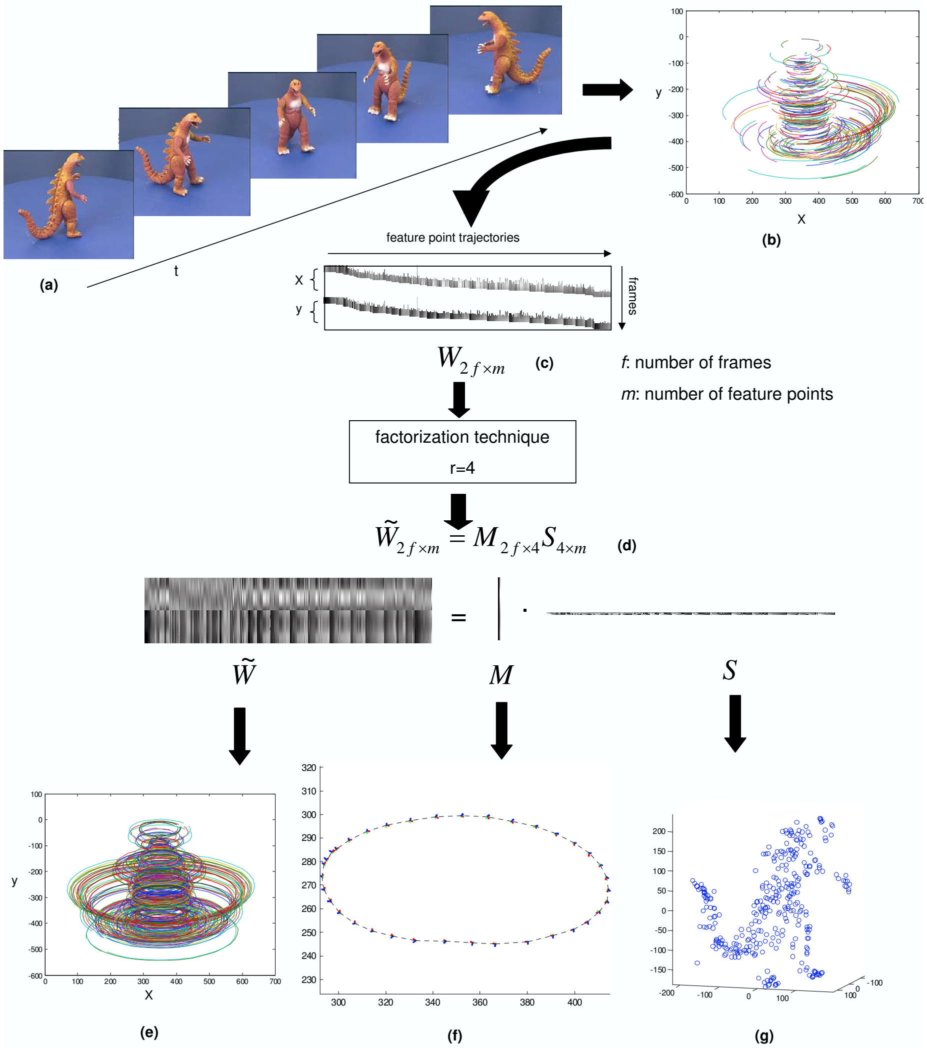


Figure 1.2: (a) five frames of a video sequence; (b) feature point trajectories plotted into the image plane; (c) matrix of trajectories W; (d) obtained decomposition by applying factorization; (e) filled in feature point trajectories, plotted into the image plane; (f) relative camera object motion, M; (g) 3D positions of the feature points, S.

contexts of photometric stereo, bioinformatics and recommender systems are proposed.

1.3 Thesis Outline

The remainder of this dissertation is organized as follows. Chapter 2 introduces the problem formulation, both for the single and the multiple objects cases. Furthermore, a detailed State of the Art of different factorization approaches focused on the SFM problem, is provided.

Chapter 3 presents the Alternation technique in detail. Due to its simplicity and good performance, this technique is used along all the thesis to obtain the best low-rank approximation of a given matrix. Furthermore, different missing data structures are presented and the performance of Alternation is studied in each case. Finally, it is empirically shown the *filtering* capability of this technique.

Chapter 4 focus on the SFM problem of a single object through factorization of matrices with a high percentage of missing data. First, a variant of Alternation focused on the SFM problem is proposed: *Alternation with motion constraints*. It consists in imposing the orthonormality of the camera axes at each iteration and the goal is to obtain an Euclidean 3D reconstruction, instead of an affine one. Additionally, it is studied the possibility of adding prior knowledge to the problem. In particular, *the smooth camera motion* is considered. An iterative multiresolution scheme is proposed in order to factorize matrices with a high percentage of missing data. The key point is to consider submatrices with a reduced percentage of missing data. Then, missing data in these submatrices are filled in by using *Alternation with motion constraints*. It is shown that the results obtained when a factorization technique is applied to the matrix filled in with the proposed iterative scheme are better than when applied directly to the given input matrix, which has a high percentage of missing data.

The SFM in the multiple object case with missing data is studied in Chapter 5. A rank estimation technique is proposed for the case of missing data matrices whose trajectories belong to multiple objects. The intuition behind the proposed technique is that, since feature point trajectories belong to surfaces of rigid objects, the behaviour of the recovered missing entries due to the movement should be similar to the one of the initially known entries. If this movement is interpreted as signal, that movement similarity is identified with the fact that the *frequency spectra* of the input matrix should be preserved after recovering missing data. Different rank values are considered and a factorization technique is applied to fill in missing entries. A measure of goodness, which compares the frequency spectra of each filled in matrix with the one of the input matrix, is defined. It is empirically shown the viability of the proposed technique. Then, any motion segmentation approach can be applied in order to segment the trajectories into different objects. Experimental results show the significance of a correct rank estimation before applying a motion segmentation technique. Finally, the SFM of every object is obtained by applying any technique for the single object case.

Chapter 6 contains several examples of applications that can be tackled with the Alternation technique apart from the Structure from Motion (SFM) problem. First of all, problems in which data can be stored into a matrix and that can present missing data are selected. Then, each problem is studied, searching for similarities and differences with the SFM problem. Finally, the Alternation is adapted to address the particular problem, if possible. The first application consists in adapting the Alternation technique to the photometric stereo problem. The goal is to recover the surface normals and reflectance of an object by using intensity images obtained under different lighting conditions. The entries corresponding to shadows and specularities are considered as missing data. These pixels have extremely low and high image intensities and may influence wrongly the final result. In a different framework, the Alternation technique is adapted to fill in missing entries in gene expression data matrices, which are generated with information provided by *DNA microarrays*. This is a technique widely used nowadays, since the information of life of any living organism resides in the function of thousands of genes. These data can provide fundamental understanding of life on the molecular level and may be useful in medical diagnosis, treatment and drug design. Finally, an adaptation of the Alternation technique is used in recommender systems, which are widely considered on *Electronic commerce*. The main objective of recommender systems is to help customers to find products they will probably like or dislike. The Alternation technique is adapted to give recommendations to customers by using information stored in the system. This information consists of rates provided by the current customer and also by all the customers that have previously used the system. For each application, experimental results, which include real data, are provided in order to show the viability of each proposed adaptation.

Finally, Chapter 7 summarizes the entire work, highlighting its principal contributions and proposing future lines of research.

Chapter 2

Factorization. State of the Art

This Chapter introduces factorization problem formulation. Firstly, the original approach [110] is presented in order to provide an overview of the factorization origin. Just as an illustration, an example of formulation for the multiple objects case [26] is provided. Furthermore, a detailed State of the Art of factorization approaches, focused on the SFM problem, is reported.

2.1 Problem Formulation

This Section introduces factorization formulation in order to show the underlying idea of the technique focused on the SFM. Firstly, the single object case proposed in [110] is presented. Then, an example of multiple objects case formulation [26] is given.

2.1.1 Single Object Case

In the original approach [110], a single static object is observed by a purely orthographic moving camera. The idea of the factorization is to project a set of 3D feature points into the image plane and to track the obtained 2D image coordinates along various frames. Let $\mathbf{p}_j$ be the 3D object points expressed in the object or world coordinate system:

$$\mathbf{p}_j = \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}, \quad j = 1, \ldots, m \tag{2.1}$$

At each frame $i = 1 \ldots f$ they are orthographically projected onto the image plane, which is defined, in the world coordinate system, by the vectors $\mathbf{i}_i$ and $\mathbf{j}_i$. These vectors are parallel to the rows and columns of the image respectively, and the projection rays are parallel to $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$. The vector from the image origin to $\mathbf{p}_j$ is defined by $\mathbf{p}_j - \widetilde{\mathbf{C}}_i$, where $\widetilde{\mathbf{C}}_i$ is the origin of the image reference system in world coordinates (see Fig. 2.1). In the image reference system, this vector is represented as:

$$\mathtt{R}_i(\mathbf{p}_j - \widetilde{\mathbf{C}}_i) = \mathtt{R}_i\mathbf{p}_j + \mathbf{t}_i \tag{2.2}$$

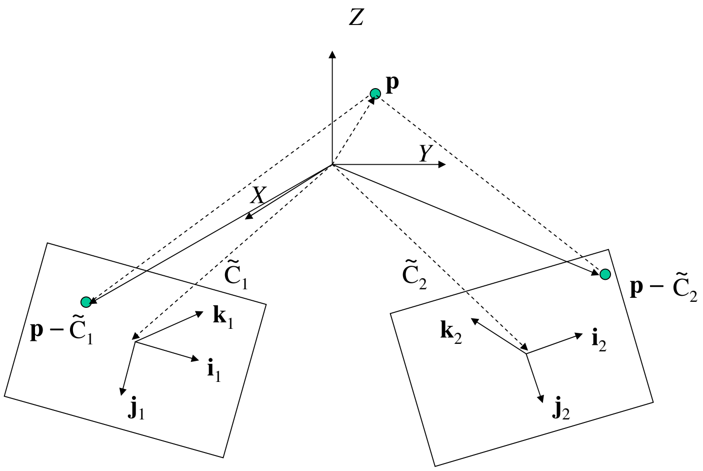


Figure 2.1: coordinate systems for the Tomasi–Kanade formulation; a static object is observed by a moving camera.

where:

$$\mathtt{R}_i = \begin{bmatrix} \mathbf{i}_i^t \\ \mathbf{j}_i^t \\ \mathbf{k}_i^t \end{bmatrix}$$

The rotation matrix $\mathtt{R}_i$ just performs a projection of the vector into the image axes, both expressed in the world coordinate system. Orthographic projection is then simply achieved by discarding the third coordinate of this $\mathbf{p}_j$ expressed in the image reference system, that is:

$$\begin{aligned} u_{ij} &= \mathbf{i}_i^t \mathbf{p}_j + t_{xi} \\ v_{ij} &= \mathbf{j}_i^t \mathbf{p}_j + t_{yi} \end{aligned} \tag{2.3}$$

Tomasi and Kanade [110] propose to get rid of the translation component in the former equations. Without loss of generality, the origin of the world coordinate system is placed at the centroid of the m feature points, so that:

$$\frac{1}{m}\sum_{j=1}^{m} \mathbf{p}_j = 0 \tag{2.4}$$

The next step consists in centering the coordinates of the features at a concrete frame. That is, in subtracting the mean of the coordinates of all the features in the same frame to each of the coordinates in that frame:

$$\begin{aligned} \tilde{u}_{ij} &= u_{ij} - \frac{1}{m}\sum_{j=1}^{m} u_{ij} \\ \tilde{v}_{ij} &= v_{ij} - \frac{1}{m}\sum_{j=1}^{m} v_{ij} \end{aligned} \tag{2.5}$$

Then, using equation (2.4), it is easy to verify that:

$$\begin{aligned} \tilde{u}_{ij} &= \mathbf{i}_i^t\mathbf{p}_j + t_{xi} - \Big(\frac{1}{m}\sum_{j=1}^{m}(\mathbf{i}_i^t\mathbf{p}_j + t_{xi})\Big) = \mathbf{i}_i^t\mathbf{p}_j \\ \tilde{v}_{ij} &= \mathbf{j}_i^t\mathbf{p}_j + t_{yi} - \Big(\frac{1}{m}\sum_{j=1}^{m}(\mathbf{j}_i^t\mathbf{p}_j + t_{yi})\Big) = \mathbf{j}_i^t\mathbf{p}_j \end{aligned} \tag{2.6}$$

The modified coordinates of the feature point trajectories are stacked into the trajectory matrix $\widetilde{W}$, where each column correspond to each of the feature trajectories and each row contains the image coordinates of all features at a given frame:

$$\widetilde{W} = \begin{bmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \dots & \tilde{u}_{1m} \\ \tilde{u}_{21} & \tilde{u}_{22} & \dots & \tilde{u}_{2m} \\ \vdots & & & \vdots \\ \tilde{u}_{f1} & \tilde{u}_{f2} & \dots & \tilde{u}_{fm} \\ \tilde{v}_{11} & \tilde{v}_{12} & \dots & \tilde{v}_{1m} \\ \tilde{v}_{21} & \tilde{v}_{22} & \dots & \tilde{v}_{2m} \\ \vdots & & & \vdots \\ \tilde{v}_{f1} & \tilde{v}_{f2} & \dots & \tilde{v}_{fm} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^t \\ \mathbf{i}_2^t \\ \vdots \\ \mathbf{i}_f^t \\ \mathbf{j}_1^t \\ \mathbf{j}_2^t \\ \vdots \\ \mathbf{j}_f^t \end{bmatrix} [\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_m] = R_{2f\times 3}S_{3\times m} \tag{2.7}$$

Notice that at this point the (slightly modified) known trajectory coordinates have been expressed as the product of two unknown matrices: the first one representing the camera motion respect to the object or world reference, and the second the 3D shape of the object. Hence, the underlying principle of the factorization technique is the affine approximation to camera imaging, as pointed out Kanatani and Sugaya [75]. It is shown later that these two matrices can be determined by taking into account the highly reduced rank of $\widetilde{W}$ and the orthonormality between certain rows of R.

Once R and S are known, the first two components of t_i, $i = 1, \dots, f$ can be readily computed from equations (2.3), (2.5) and (2.6):

$$t_{xi} = \frac{1}{m}\sum_{j=1}^{m} u_{ij} \tag{2.8}$$

$$t_{yi} = \frac{1}{m}\sum_{j=1}^{m} v_{ij} \tag{2.9}$$

Factors computation

Since R is $2f \times 3$ and S is $3 \times m$, in the absence of noise, the rank of $\widetilde{W}$ is at most 3. Cases where rank is less than 3 are called degenerated, and they occur when the shape and/or the rotation matrix are not of full rank. This happens in the following situations:

- feature points $\{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ do not span a 3D space, i.e. they are coplanar (rank 2), or collinear (rank 1)
- the camera rotation is not performed over its three degrees of freedom, for instance rotates around a certain 3D vector for all frames (rank 2), or does not rotate at all (rank 1)

As this Section is included just to introduce the simplest version of factorization, degenerated cases are not studied. Thus, it is assumed that rank($\widetilde{W}$)=3.

If $2f \geq m$ the Singular Value Decomposition (SVD) of $\widetilde{W}$ gives the following decomposition:

$$\widetilde{W} = U_{2f\times m}\Sigma_{m\times m}V^t_{m\times m} \tag{2.10}$$

where Σ is a diagonal matrix containing the singular values σ_i sorted in decreasing order, that is:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{m-1} \geq \sigma_m \tag{2.11}$$

Actually, as the rank is 3, only the first three singular values are not zero. Thus, all the rows and columns of Σ, except the first three ones, can be discarded:

$$\widetilde{W} = U_{2f\times 3}\Sigma_{3\times 3}V^t_{3\times m} \tag{2.12}$$

However, in practice, trajectories are noisy. Actually, just a very slight noise can make $\widetilde{W}$ full rank (m, since $2f \geq m$). An important fact that should be taken into account is that to nullify the 4th to m singular values is equivalent to calculate the least squares error approximation of $\widetilde{W}$, which has rank 3 (see [43] for details). Which is the same, given a noisy matrix $\widetilde{W}^* = U\Sigma V^t$, its closest rank-$r$ approximation matrix is:

$$\widetilde{W} = U_{2f\times r}\Sigma_{r\times r}V^t_{r\times m} \tag{2.13}$$

being the error:

$$||\widetilde{W}^* - \widetilde{W}||^2_F = \sigma^2_{r+1} + \ldots \sigma^2_m \tag{2.14}$$

where $\|\cdot\|_F$ is the Frobenius norm [43]. This is, in fact, the expression to minimize when a matrix is approximated by a rank-r matrix.

Therefore, singular values can be used as a measure of the noise, though in some factorization settings the issue of determining the right or best rank (when several are feasible) is an important unsolved problem.

From the above decompositions (2.7) and (2.12), it can be deduced that:

$$R = U_{2f\times 3}\Sigma^{\frac{1}{2}}, \quad S = \Sigma^{\frac{1}{2}}V^t_{3\times m} \tag{2.15}$$

However, this decomposition is not unique, since any 3×3 invertible matrix A also provides a valid decomposition $\widetilde{W} = RAA^{-1}S$. Then:

$$\widetilde{W} = \hat{R}\hat{S} \tag{2.16}$$

where

$$\hat{R} = RA \quad \hat{S} = A^{-1}S \tag{2.17}$$

being R and S the true rotation and shape matrices. The matrix A is computed by imposing orthonormality between the two camera axes at each frame. That is:

$$\hat{R}\hat{R}^t = I = RAA^{-1}R^t \tag{2.18}$$

being I the identity matrix.

And the following $3f$ non–linear equations on the terms of A hold:

$$\begin{array}{rcl} \mathbf{i}_i^t AA^{-1}\mathbf{i}_i & = & 1 \\ \mathbf{i}_i^t AA^{-1}\mathbf{j}_i & = & 0 \\ \mathbf{j}_i^t AA^{-1}\mathbf{j}_i & = & 1 \quad i = 1, \ldots, f \end{array} \tag{2.19}$$

This final step is referred to as *normalization*. According to [110], this is a simple data fitting problem which, though non–linear, can be solved efficiently and reliably.

Another way to solve it, is to calculate the 6 elements of the symmetric matrix $X = AA^{-1}$, converting the former equations into a linear system of the type $\mathbf{Bx} = c$ where $\mathbf{x} = [X_{11}X_{12}X_{13}X_{22}X_{23}X_{33}]^t$. An analogous system is solved in Costeira and Kanade's approach [26], introduced in next Section, and also in Han and Kanade's algorithm [53].

Before going into details, it should be noticed that the solution is determined up to a rotation, since only the position of the world reference system has been imposed. One can fix its orientation by representing the different coordinate systems relative to that of the first frame: $\mathtt{R}_i' = \mathtt{R}_1^{-1}\mathtt{R}_i, S' = \mathtt{R}_1 S$, $i = 1, \ldots, f$.

2.1.2 Multiple Objects

The seminal approach presented by Tomasi and Kanade [110] is formulated for a single static rigid object viewed by a moving camera. Considering scenes with multiple objects, this approach could be applied to each of the objects separately and the SFM of each of them could be obtained. In general, when working with multiple objects, the trajectories of features corresponding to the same object are not sorted accordingly. Hence, some kind of segmentation should be applied in order to group trajectories belonging to the same object. To obtain the motion segmentation, it should be used the fact that under affine camera model, trajectories corresponding to the same object lie in the same linear subspace. The goal is to find each of these linear subspaces and to reduce the matrix of trajectories to a form that enhances them. Unfortunately, the dimensionality and dependency of these linear subspaces are unknown. Once the clusters are obtained, any single object technique can be used to obtain the shape and motion of each of the objects.

As explained later in Chapter 5, when there are several objects in the scene, an additional problem should be considered: the estimation of the rank of the matrix of trajectories. In the case of a single rigid moving object, the rank of the matrix of trajectories W is at most 4 (3 if the translation component is subtracted). This rank value is used in most of the single object case approaches. In the multiple objects case, since both the number of objects in the scene and the kind of their movement are unknown, the rank of the matrix of trajectories can not be known a priori. Furthermore, due to the noise in the data, if the rank is directly computed, it is usually the number of rows or columns of W. Therefore, in the multiple objects case, most approaches estimate first the rank of the matrix of trajectories, which depends on the noise in the data.

This Section presents the formulation proposed by Costeira and Kanade [26], in order to illustrate the multiple objects case. The authors suppose two or more independently moving objects viewed by a static camera. In fact, the motion of observed objects is relative to the camera; objects and camera can move freely in 3D space.

Let $\mathbf{p}_j$, $j = 1, \dots, m$ be the 3D object points expressed in some arbitrary coordinate system fixed to the object. At each frame $i = 1 \dots f$, these points are projected into the image according to the simple orthographic camera model:

$$u_{ij} = \mathbf{i}_i^t \mathbf{p}_j + t_{xi} \tag{2.20}$$

$$v_{ij} = \mathbf{j}_i^t \mathbf{p}_j + t_{yi} \tag{2.21}$$

being $\mathbf{i}_i, \mathbf{j}_i$ the camera axes and (t_{xi}, t_{yi}) its translation. Notice that these equations are identical to equations (2.3).

As in the original approach, all image trajectories (u_{ij}, v_{ij}) are stacked into a $2f \times m$ measurement matrix W, which can be decomposed as the product of the motion and shape matrices:

$$\begin{bmatrix} u_{11} & \dots & u_{1m} \\ \vdots & & \vdots \\ u_{n1} & \dots & u_{nm} \\ v_{11} & \dots & v_{1m} \\ \vdots & & \vdots \\ v_{n1} & \dots & v_{nm} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^t & t_{x1} \\ \vdots & \vdots \\ \mathbf{i}_n^t & t_{xn} \\ \mathbf{j}_1^t & t_{y1} \\ \vdots & \vdots \\ \mathbf{j}_n^t & t_{yn} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_m \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$$W_{2f\times m} = \begin{bmatrix} R_{2f\times 3} & \mathbf{t}_{2f\times 1} \end{bmatrix} S_{4\times m} = M_{2f\times 4} S_{4\times m} \tag{2.22}$$

The translational component of motions can not be eliminated by placing the coordinate system at the mass center of feature points, as in the seminal approach [110]. That is due to the fact that the center mass of each object can move differently. Hence, in the absence of noise, in this case the rank of the matrix of trajectories W is at most 4, instead of 3. The SVD decomposition $W = U_{2f\times 4}\Sigma_{4\times 4}V_{4\times m}^t$ yields the true M and S but for an unknown invertible matrix $A_{4\times 4}$: $W = MS = \hat{M}AA^{-1}\hat{S}$, with $\hat{M} = U\Sigma^{\frac{1}{2}}$, $\hat{S} = \Sigma^{\frac{1}{2}}S$. Fortunately, A can be computed through the aforementioned *normalization* process by using the fact that the rows of M represent the camera rotation axes and satisfy, for $i = 1, \ldots, f$, the following conditions:

$$\begin{aligned} \mathbf{i}_i^t A A^{-1} \mathbf{i}_i &= 1 \\ \mathbf{j}_i^t A A^{-1} \mathbf{j}_i &= 1 \\ \mathbf{i}_i^t A A^{-1} \mathbf{j}_i &= 0 \end{aligned} \tag{2.23}$$

For simplicity, the formulation for the case of only two objects is presented. The n-objects case can be deduced analogously. Suppose that the scene contains two objects moving independently. Suppose also that the trajectories corresponding to each object are known. Then, their corresponding trajectory matrices, W_1 and W_2, can be constructed. Then, the matrix of all trajectories sorted by object can be represented as:

$$W^* = [W_1 | W_2] \tag{2.24}$$

It can be easily seen that W^* factorizes as before into the product of two block matrices M^* and S^*, each one containing the motion and shape submatrices of each object:

$$\begin{aligned} W^* &= [M_1 | M_2] \begin{bmatrix} S_1 & \mathbf{0} \\ \mathbf{0} & S_2 \end{bmatrix} \\ &= [\hat{M}_1 | \hat{M}_2] \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix} \begin{bmatrix} A_1^{-1} & \mathbf{0} \\ \mathbf{0} & A_2^{-1} \end{bmatrix} \begin{bmatrix} \hat{S}_1 & \mathbf{0} \\ \mathbf{0} & \hat{S}_2 \end{bmatrix} \\ &= \hat{M}^* A^* A^{*-1} \hat{S}^* \end{aligned} \tag{2.25}$$

The single object case algorithm could be extended to find out the linear transformation A^* with the structure of equation (2.25).

Hence, if the scene contains p objects moving independently, each with a certain number of features, the matrix of all trajectories sorted by object W^* factorizes as before into the product of p block matrices M_k and S_k, $k = 1, \ldots, p$. Unfortunately, in most cases the matrix W^* is not known, since features are not sorted by object. A matrix W, which is equal to W^* but having permuted some columns, should be considered.

Therefore, the SFM problem in the multiple objects case should tackle first a preprocessing step that segment the features of the scene into the different objects. Once the segmentation is achieved, single object techniques can be applied to each of the objects in order to obtain the shape and relative motion of each of the objects.

Costeira and Kanade [26] propose the permutations of the columns of W until the matrix W^* is obtained. On of the main contributions of their method is the so called shape interaction matrix, defined as follows:

$$Q^*_{m \times m} = V^* V^{*t} \tag{2.26}$$

Each row (or column) of the matrix Q^* represents each of the feature point trajectories. This symmetric matrix has an interesting block–diagonal structure: if features i and j belong to different objects, $Q(i,j)^* = 0$. Hence, the diagonal of this matrix must be equal to 1. This property holds on even though the trajectories are not sorted by objects. Therefore, since W is equal to W^* but having permuted some columns, so V^t is equal to V^{*t} only that permuting the same set of columns. Consequently, Q^* will result by permuting rows and columns of Q in the same way.

The Structure from Motion problem for multiple moving objects has been reduced to find out the right row and column permutations of Q such that it becomes block–diagonal; then, the single body algorithm can be applied. However, recovering Q^* from Q, and thus performing feature grouping (motion segmentation), is still a tough problem because noise makes the entries of Q corresponding to features of different objects not exactly zero. Furthermore, an important restriction of this approach, which reduces its application, is that the objects must move independently. More details on the algorithm to obtain the matrix Q^* and also to find its blocks can be found in [26].

2.2 State of the Art

Factorization approaches that tackle the SFM problem are summarized in this Section. Its structure is based on the distinction between the single and multiple objects cases and also between the full and missing data cases.

2.2.1 Single Object, Full Data Case

The already presented original formulation [110] considers a static object observed by a moving camera. As detailed above, the origin of the world coordinate system is placed at the centroid of all the feature points and the translation component is eliminated. Thus, in the absence of noise, the rank of the matrix of trajectories (which contains the image coordinates of the points) is at most 3. Tomasi and Kanade consider the simplest affine camera model: the orthographic projection. Many other approaches take advantage of the simplicity of this camera model.

However, since the camera projections are only well approximated by affine mappings when the distance of the object to the camera is greater than the depth variation of this object, other camera models are also introduced. Details on the different camera models can be found in [55]. In [91], Poelman and Kanade consider the parapersepective model and show how it approximates closely the perspective projection, while retaining linear algebraic properties of the orthographic projection. Concretely, the paraperspective model accounts for the scaling effect of an object as it moves towards and away from the camera, as well as the different angle from which an object is viewed as it moves in a direction parallel to the image plane. As in the original approach, the translation vector is subtracted from the matrix of trajectories and the rank of the resulting matrix is 3 again. Although the factorization process is similar to the one in the orthographic case, with this camera model, the depth (which is actually not known) should be taken into account in the normalization step. Due to that fact, the algorithm becomes more complicated. Additionally, Poelman and Kanade present an iterative method to recover the shape and motion by using a perspective projection model. The problem is formulated as a non-linear least squares problem in the motion and shape variables. As in any iterative method on non-linear minimization, the final result depends on the initialization. Poelman and Kanade propose to take the shape and motion provided by paraperspective factorization and refine them alternately.

Factorization methods for perspective images have also been proposed. With the perspective camera model, and without calibration nor metric information of the scene, a projective reconstruction is obtained. In order to obtain it properly, a set of projective depths must be recovered. That is, a set of projective scale factors that represent the depth information lost during the image projection. In the affine case, these scales are constant and are not taken into account. Sturm and Triggs [106] propose a method for recovering these projective scales, using only fundamental matrices and epipoles estimated from the image data. Given two images, the fundamental matrix and the epipoles establish a relation between the projective depths of a single 3D point in these two images. The number of fundamental matrices and epipoles needed to obtain a complete set of projective depths is not especified in [106]. In theory, the more fundamental matrices are available, the better is the obtained solution. Unfortunately, computing a high number of fundamental matrices and epipoles turns the problem computationally very expensive. Once the image coordinates are rescaled (with the computed projective depths) a measurement matrix of rank 4 is obtained and the singular value decomposition (SVD) is applied to factorize it. Triggs presents in [113] a similar approach, which also uses fundamental matrices and epipoles. The main contribution is the extension of the previous method to use not only 3D points but also 3D lines to construct the matrix of trajectories. It is concluded that line reconstruction is less stable, especially due to the presence of outliers when the cameras are collinear or coplanar. Besides, Triggs proposes to use an approximate fixed-rank factorization technique which runs faster than the SVD in large problems.

As pointed out in [57], the main drawback of these two algorithms is that they can only deal with 2, 3 or 4 images and there is no generic way to extend them to more images. Heyden et al. [57] introduce an iterative factorization algorithm for projective structure and motion. The idea of this algorithm is to begin with a set of only a few images and to add images in the following steps in order to update the reconstruction and the relative depths. Their algorithm relies on subspace formulation and one of the main advantages is that it is independent of the chosen coordinate system in the image as well as of the ordering of the points.

Christy and Horaud [25] describe a method for solving the Euclidean reconstruction problem with a perspective calibrated camera. The idea is to incrementally perform Euclidean reconstruction with either a weak or a paraperspective camera model. Therefore, at each iteration, an affine reconstruction is obtained, which converges to a Euclidean shape and motion parameters that are consistent with a perspective model. They define *perspective corrections*, which are modified at each iteration. This method converges in a few iterations and solves the sign ambiguity that give the affine reconstruction. As pointed out by the authors, the method fails to converge when there are feature points very close to the camera.

As in the aforementioned paper [113], other approaches consider features different from the points to obtain the data. In [93], Quan and Kanade present an approach in which the shape and motion are recovered from line correspondences with the affine camera instead of points. The authors propose a multi-step factorization method based on the decomposition of the shape and motion into three separate substructures, which can be linearly solved by factorizing the appropriate matrix of trajectories. In a different approach, Aguiar and Moura [3] present a *surface-based factorization* method which consists in tracking regions instead of tracking a large number of features. Firstly, the unknown shape of the 3D rigid object is described by using polynomial patches. Then, it is used the fact that the projections of these patches in the image plane move according to parametric 2D motion models. Finally, the parameters which describe the 3D shape and motion from the 2D motion parameters are recovered by factorizing rank-1 matrices. Although this proposed approach is derived from the orthographic projection model, it can be extended to the scaled-orthographic and the paraperspective projections.

Morris and Kanade [87] propose a unified factorization algorithm for the SFM recovery by using feature points, line segments and planes. The authors aim to define an algorithm robust to the imprecision and small errors involved in image registration. Their algorithm consists in adding prior information on the coplanarity of features which provide additionally constraints for the shape recovery. A weighted least squares motion and shape recovery problem is proposed. Directional uncertainty measures for feature location are incorporated. In addition, an uncertainty measure for resulting shape features is provided. Kahl and Heyden [70] present several methods for the Structure from Motion estimation under affine camera model that use points, lines and conic features. They propose three new tensor representations for matching constraints approaches. However, these representations are limited only to a few views. A factorization technique that uses the previous three-views tensor constraints and that can handle arbitrarily views is also introduced. Furthermore, a method based on the closure constraints [114] to handle missing data is proposed. One of the main problems working with matching constraints is to obtain feature correspondences between different images. In [70], these correspondences are assumed to be known.

In [6], Anandan and Irani rightly pointed out that SVD is a global solution to the factorization problem only when the x and y positional errors in the features are uncorrelated and identically distributed. Unfortunately, this is not always true in real data. Uncertainty in feature position depends on the underlying spatial intensity structure in the image, which has strong directionality to it. They present a new factorization approach that minimizes the covariance-weighted squared-error. It consists in transforming the data into a covariance-weighted data space, where the components of noise in the different directions are uncorrelated and identically distributed. Therefore, the SVD can be applied to the transformed data. In a linear but suboptimal second step, the shape and motion of the original data space are obtained. Noisy feature correspondences with high degree of directional uncertainty can be factorized into shape and motion. This method allows to treat corner-like points together with points along linear structures in the image.

Aanaes and Fisker [1] present a factorization algorithm that aims at effectively deal with errors in the tracked features: mismatched features, missing features and noise on the individual features. Their proposed algorithm enables the use of robust statistical techniques and arbitrary noise models for the individual features. It is based on the aforementioned Christy-Horaud's factorization scheme [25]. The main idea of the proposed algorithm is to use a robust error function which is implemented via Iteratively Reweighted Least Squares (IRLS). It is an iterative method that fits the model to the data by minimizing the weighted least squares of the residuals (see [100] and [52] for more details on IRLS).

In [2], and more extensively in [4], Aguiar and Moura describe a rank 1 weighted factorization method that uses the orthographic camera model. The weights are defined by the inverse of the estimates error standard deviation so that the $2D$ motion estimates for sharper features (which are usually well-estimated) are given more weight, while the noisier motion estimates for smoother textures are weighted less. Aguiar and Moura [4] are the first in exploiting the freedom to choose the relative alignment between the object and camera coordinate systems in one of the images in the sequence and develop a method that recovers the SFM through the factorization of a rank 1 matrix.

A different approach is introduced in [45] by Gruber and Weiss. As in [6], they pointed out that in real situations the observed data have different directional uncertainty. However, most of the proposed factorization techniques do not take it into account. They propose to model the noise by a covariance matrix and to use the expectation-maximization (EM) algorithm to obtain the factorization. Concretely, the structure S whose form depends on the structure of the defined covariance matrix, is recovered in the M-step of the algorithm. In addition, Gruber and Weiss propose to add a prior on the temporal trajectory of the camera position. Namely, they use the fact that the camera location at time $t + 1$ is probably similar to its location at time t. They show how the performance of the factorization is improved by adding this temporal prior. As commented in next Section, this approach can deal also with missing data in the matrix of trajectories.

Huynh and Heyden [61] present a factorization approach under affine projection that detect and eliminate outliers from the matrix of trajectories. They consider subspaces spanned by different subsets of columns in the matrix of trajectories and introduce a similarity measure on these subspaces. This measure is used to detect and eliminate the outliers in the observations. Only the inliers observations are used to factorize the matrix of trajectories. In a posterior paper [62], the same authors study the relationship between subspace distances and reprojection errors. They pointed out that outliers are image features that do not conform to the projection of the true structure. Therefore, when the true structure is back projected onto the image plane, outliers show up with large reprojection errors. Two algorithms for computing reprojection errors for outlier detection are proposed. More recently, Huynh et al. [60] present an outlier correction scheme that iteratively updates the elements of the matrix of trajectories. Thus, the method corrects the outliers and factorizes the matrix of trajectories simultaneously. In this iterative scheme the *Alternation* method, presented in next Section, is used to compute the shape and motion factors instead of the SVD. Its computational cost is smaller than the one of the SVD and in addition, it can handle missing entries in the matrix of trajectories, as explained in next Section.

As pointed out in [86], the factorization technique is based on a batch-type computation; that is, it recovers shape and motion after all the input images are given. Due to that fact, it is limited to off-line applications. Besides, it needs to store a large matrix whose dimensions increase with the number of frames. In order to be able of adding data to the matrix of trajectories, sequential algorithms have been proposed. These methods avoid to work with large amount of missing data and allow to work sequentially, which is a very important fact in real-time applications. At the same time, they take advantage of previous measurements. Morita and Kanade [86] presents a sequential algorithm which gives estimates of shape and motion at each input frame. A covariance-like matrix is stored, instead of feature points, and its size remains constant as the number of frames increases. The *orthogonal iteration method* [43] is used to update the three dominant eigenvectors instead of the SVD. A different recursive factorization method is proposed in [80]. The shape space is estimated within a mean square error (MSE) minimization framework. This method has both a very low complexity and good performance which make it suitable for real-time applications.

2.2.2 Single Object, Missing Data Case

In real situations, features can disappear due to object self-occlusions, or because they are not more visible in the sequence, or simply due to failures of the tracker. These situations give uncomplete trajectories. Several factorization approaches have been proposed to deal with missing data in the trajectories. The SVD can not be used when there are missing data and other methods to decompose the matrix of trajectories into the shape and motion factors should be introduced.

In the seminal approach, Tomasi and Kanade [110] propose an initialization method in which they first decompose the largest full submatrix by the factorization method and then the initial solution grows by one row or by one column at a time, completing thus the missing data. The problem is that finding the largest full submatrix is a NP-hard problem. Moreover, the solution depends on the order in which the columns or rows are completed.

In a completely different approach, Jacobs [66] treats each column with missing entries as an affine subspace and shows that for every r-tuple of columns, the space spanned by all possible completions of them must contain the column space of the completely filled matrix. Unknown entries are recovered by finding the least squares regression onto that subspace. In order to obtain a good result, all the r-tuple affine subspaces should be used. However, in practice, only a selection of them is considered, which can give undesirable results. In addition, this algorithm is strongly affected by noise on the data. Nevertheless, it gives a useful initial guess for iterative methods.

Chen and Suter [24] provide an iterative algorithm to recover the missing components in a large noisy low-rank matrix. The algorithm begins with a complete submatrix which grows at each iteration by one row or column, filling in the missing entries at the same time. They present a criterion based on the SVD's *denoising* capability versus missing data in order to decide which parts of the matrix should be used in the iterative imputation/recovery process. Given an incomplete matrix, they define its *unreliability* as the ratio between the number of its independent variables and the number of nonmissing components. The goal is to recover the most *reliable* incomplete submatrix by using the iterative algorithm. Then, the other columns and rows are projected onto it by using an imputation method. Focusing on the SFM problem and in order to find the submatrix which has the minimal u*nreliability* ratio, the vectors which has the least nonmissing components are excluded, until the *unreliability* ratio begins to increase.

In [67], Jia et al. present an algorithm that aims the SFM recovery with noisy and missing data. It is similar to the aforementioned one [66]. However, the authors point out that the randomness in selecting the submatrices in [66] cannot guarantee a consistently accurate recovery of missing data. Hence, instead of selecting several triple columns, they propose to use the most reliable submatrix to recover the 3D structure. They define a criterion that provides a measure of the sensitivity of a submatrix to noise perturbation: the deviation parameter (DP). Using this criterion, the submatrices with smallest deviation parameter are considered to construct the final matrix. Then, the SVD of this matrix is computed. Since the DP criterion can not guarantee the optimal selection of submatrices, the algorithm is implemented in n different subsets of the matrix and the average of all the results gives the solution. The performance of this approach is compared to the Jacobs' one [66] and to the Chen and Suter's one [24]. Jia et al. criticize the fact that with this last approach, the rows/columns that have more known entries are in general selected, which not always give the best solution.

Wiberg [123] presents an algorithm that uses the Gauss-Newton method to compute the principal components of a matrix of data with missing observations. The key point is to separate the variables in two sets and compute them alternatively. Based on that, Shum et al. [104] propose a method for 3D object modeling. In a recent approach, Okatani and Deguchi [89] present in detail Wiberg's algorithm focusing on the matrix factorization problem and demonstrating its performance compared to the Levengerg-Marquardt (LM) technique.

Wiberg's algorithm is generally referenced in the literature (e.g., [23], [50]), as the origin of what is called the *Alternation* technique. This approach uses the fact that if one of the factors (A or B) is known, the other one can be computed by solving a least squares problem. Concretely, the algorithm starts with an initial random matrix (A or B) and computes alternatively each of the factors until the product AB converges to W. Several variants of this method have been presented in the literature. Since this technique is widely used in this thesis, a detailed study can be found in Chapter 3.

In [50], and previously in [49] and [48], Guerreiro and Aguiar introduce the *Row-Column* algorithm, which is very similar to the *Alternation* algorithm. They study its performance and compared it with the Expectation-Maximization (EM) algorithm. They conclude that it performs better than the EM. Furthermore, it is more robust to the initialization. Hartley and Schaffalitzky [54] suggest adding a normalization step between the two factors updated at each iteration. This particular Alternation technique is denoted as *PowerFactorization*. Furthermore, the authors propose another variant to Alternation, focusing on the SFM problem. In this case, A and B factors correspond to the motion and shape matrices, respectively. Hence, since B contains the 3D feature points in homogeneous coordinates, it can be imposed that the last row of B is equal to $\mathbf{1}$ (where $\mathbf{1}$ represents a vector of 1). Roweis [95] define an Alternation approach within the Expectation-Maximization (EM) framework for principal component analysis (PCA).

The papers [60] and [1] presented in Section 2.2.1 can also deal with missing data and both use the *Alternation* technique to factorize the matrix.

In [23], and more extended in [22], a report of the variants of Alternation, including their algorithm, is given. Buchanan and Fitzgibbon [23] propose the *Alter-*

nation/Damped Newton Hybrid, which combines the Alternation strategy with the *Damped Newton* method. The latter is fast in valleys, but not effective when far from the minima. The goal of introducing this hybrid scheme is to give a method that has fast initial convergence and, at the same time, has the power of non-linear optimization. However, they conclude that second order non-linear optimization strategies performs better than the hybrid schemes.

Gonçalves and Aguiar [44] propose a method to decide wether a region that has become visible is a region that has been seen before. A global cost function that balances the model fidelity and complex penalization is defined. The goal is to compute the simplest 3D rigid object that best matches the entire set of 2D image projections. Their method shows that the obtained 3D reconstruction is more accurate when re-appearing regions are taken into account.

In [83], Martinec and Pajdla present a method to obtain the projective shape and motion by factorization. The method proposed in [106] is used to estimate the projective depths of image points, while missing data are filled in by using an extension of the method proposed in [66]. The main idea of their proposed method consists of computing first the projective depths of all known points in the matrix of trajectories. Then, its missing entries are filled in by finding a complete matrix of rank 4 as close as possible to it. This method provides a unique solution that can be used as initialization of non-linear optimization such as bundle adjustment. Furthermore, it can deal with outliers if they are few compared to the number of inliers.

Koh [108] proposes a method for estimating 3D positions of feature points that is robust to the sensibility to noise. A comparison of this robustness to the approaches [66] and [50] is provided. The key point of that approach is to use geometrical correlations between a 2D observation matrix and 3D shape.

Techniques for recovering the Structure from Motion with missing data that are not purely factorization techniques have also been proposed. Some of them apply a factorization technique in internal steps. Below, three of them are presented.

In [18], Brandt presents a closed-form maximum likelihood solution for the translation vectors and 3D coordinates of the tracked features points under missing data. All feature points are used simultaneously. The author [18] formulates the problem in a different way of previous factorization approaches. A cost function is defined and the goal is to find the least-squares solution for the parameters that minimizes it. It is pointed out how the translation parameter values must be determined after computing the reconstruction in contrast to the case when there are not missing data. An iterative algorithm for affine reconstruction is proposed: first the Jacobs' method [66] is used to obtain an initialization and then, an expectation maximization (EM) algorithm gives the maximum likelihood estimation under missing data. Other applications of this solution are that it can be used as an initial solution for the 3D coordinates in bundle adjustment and also to identify degenerate image configurations. Brandt compares the method to Jacobs's [66] and Shum et al.'s [104] approaches. Additionally, with synthetic data, the methods are also compared to factorization, using only points visible in all the views.

A technique for 3D reconstruction by fitting low-rank matrices with missing data is proposed in [84]. It consists in taking rank-four matrices of minimal size and in combining spans of their columns in order to constraint a basis of the whole fitted matrix. The solution is valid for the affine and the perspective camera models and for scenes of various types: open and closed sequences in both narrow and wide baseline setups. This method does not try to fill in the missing data. In fact, only the known data is used. The formulation is similar to the one presented in [66]. The main difference is that the problem is formulated in terms of the original subspaces, while in [66] the complementary ones are used.

Guilbert et al. [51] present a batch method from recovering Euclidian structure and motion form sparse image data. Using closure constraints [114], the camera coefficients are formulated linearly in the entries of the affine fundamental matrices. An important characteristic of the algorithm is that many constraints can be easily imposed: camera position or orientation in a particular frame, equality or proximity between consecutive camera poses, equality of 3D points, smoothness of camera trajectory, etc.

2.2.3 Multiple Objects, Full Data Case

This Section reports a review the state of the art in the SFM when multiple objects and full matrices of trajectories are considered. Some of the presented approaches focused the attention on the motion segmentation step. Once the trajectories are clustered into different objects, any single object technique can be applied to obtain the shape and motion of each of the objects.

Boult and Brown [12] propose to use the SVD of a matrix of trajectories to obtain the segmentation of multiple motions. Their proposal is justified by the relation between the SVD and the principal component analysis (PCA). Concretely, they use the right singular vectors associated with the nonzero singular values to obtain the motion segmenation. The problem is that due to the noise in the input data, all the singular values are in general nonzero and a strategy to estimate the number of nonzero singular values or, which is the same, the rank of the input matrix, is needed to be estimated. Firstly they define an upper and a lower bound on the rank value and secondly the logarithm of singular values beyond these bounds is computed. The rank is approximated as the one for which the logarithm achieves maximal curvature.

In [26], Costeira and Kanade define the interaction matrix computed from the SVD of the input matrix. This matrix gives the correct motion segmentation using the zero product between independent trajectories. Concretely, entries corresponding to features from different objects are zero and nonzero otherwise. This property gives a block-diagonal structure to this matrix. One of the main drawbacks of this technique is that the noise in the data is propagated to the interaction matrix and entries from features of different objects may not be zero. Furthermore, the columns of the interaction matrix must be sorted in order to obtain the desired block-diagonal structure. Hence, all the possible rank values should be tested and the one that gives the best block-diagonal structure of the interaction matrix is selected.

A similar formulation is presented by Han and Kanade in [53]. It is assumed that the objects move linearly with constant speeds. In this case, not only the shape and motion are obtained, but also the velocity of each of the feature points. One advantage of this approach in front of the previous one is that it is not necessary to segment the features before obtaining the shape and motion. In fact, the segmentation is obtained by grouping the recovered velocity values of the feature points. The rank is at most 6 with this formulation (smaller in degenerate cases). The main drawback of this technique is that, in order to obtain the correct velocity values, the static points should be known and their relative velocity (notice that the camera is also moving) should be subtracted to all the recovered velocities. Unfortunately, without previous knowledge or assumptions, the static points are not known.

In a different approach, Ichimura and Tomita [64] introduce a method for motion segmentation based on feature selection. The idea is to construct a basis of linear space that represents the shape of the objects. The interaction matrix defined by Costeira and Kanade [26] is used. Ichimura and Tomita propose to apply the SVD and to estimate the rank of the matrix of trajectories r by studying the obtained singular values. Then, the QR decomposition [43] of the shape interaction matrix is used to select r features and to obtain the basis vectors of the shape space, which give the number of objects and their segmentation. Finally, the remaining features are also segmented by using the orthogonal projection matrix for each shape space (defined by the basis vectors). However, if data are noisy, it will be difficult to estimate the rank of the matrix properly.

Ichimura [63] proposes an algorithm for motion segmentation which also uses the orthogonal projection matrix of the shape space. As before, the interaction matrix defined in [26] is used as orthogonal projection matrix of the shape space. It is obtained by computing the SVD of the matrix of trajectories. In order to introduce robustness in the case of noisy data and outliers, a method for feature selection using *discriminant criterion* is presented. The idea is to select the feature with the most useful information for segmentation. Then, a group of features is extracted based on the result of discriminant analysis for the selected feature. The same procedure is applied recursively to remaining features to extract other groups. The rank of the matrix of trajectories is needed to be known to apply the SVD. However, it is difficult to estimate with the presence of noise or outliers in the data. The rank corresponding to the segmentation result with maximum separation of entities of orthogonal projection is selected.

Kanatani [72] reformulates the Costeira and Kanade [26] approach by defining an interaction matrix computed from the original data. The geometric information criterion (AIC) defined in [71] is used as a measure of goodness of the model. Concretely, the AIC evaluates if two groups of points should be or not merged. One disadvantage of the approach presented in [72] is that the number of independent motions is assumed to be only two: one corresponding to the background and one to a moving object.

In a more recent paper, Kanatani and Matsunaga [74] propose an approach that also uses the geometric AIC to estimate the number of independent motions for multiple objects motion segmentation, when data are free of noise. At the same time, the rank of the matrix of trajectories is estimated. A possibility for estimating the rank of a set of vectors is to fit them with subspaces of different dimensions and adopt the dimension of the subspace having the smallest residual (the sum of the square distances of the data points to the fitted subspace). However, the higher the dimension of the subspace is, the smaller the residual becomes. The idea of the AIC is to balance the increase of the residual against the decrease of the degree of freedom.

More details can be found in [71]. In [73], Kanatani proposes a strategy that also uses the geometric AIC to estimate the rank in the case of noise in the data.

One disadvantage of all the above techniques is that they are valid only for independent motions. In [130], Zelnik-Manor and Irani analyze the degeneracies in multiple objects and multi-sequence factorization and present an approach to segment objects in such cases. They suggest to construct an affinity matrix by using only the most dominant of their eigenvectors instead of taking all them. In order to approximate the rank of the matrix of trajectories, the ratio between the sorted singular values is studied.

Sugaya and Kanatani [107] study the geometric structure of the degeneracy of the motion model. Concretely, the authors studied what they call the *parallel 2D plane degeneracy*. Sugaya and Kanatani point out that this is a common situation in real sequences. The key point of their proposal is to use first the degenerate motion model and then the general 3D motion model. A comparison of the performance of their proposed approach to other motion segmentation techniques is provided in their paper.

Zelnik-Manor et al. [132], explore the meaning of applying multi-body factorization to the matrix obtained by stacking the x and y coordinates of the 2D trajectories horizontally. In most factorization approaches, the coordinates are stacked vertically. That is, each column corresponds to each of the trajectories; the first rows corresponding to the x coordinates and the last ones to the y coordinates of the trajectories. In this novel approach, each row contains the flow of all points between a pair of frames, typically between a reference frame and one of the other frames (more details can be found in [65]). The obtained matrix is referred to as *flow-field matrix*. The authors define the *motion consistency* based on temporal constraints, which consists in grouping feature points that have a consistent behavior over time. Besides, an algorithm for segmenting the image into multiple objects is also presented. It applies this new multi-body subspace (rows instead of columns) to image brightness quantities. One of the main advantages of this approach is that it does not require prior correspondence estimation or feature tracking.

Zelnik-Manor and Irani [131] present a similar approach that consists in obtaining temporal segmentation. Instead of grouping points moving with consistent motions, frames that capture *consistent shapes* are grouped by clustering the rows of the matrix of trajectories. One advantage of the temporal factorization is that the temporal clustering of frames can be obtained with only a few tracked points. Furthermore, the dimensionality of the data is smaller than for spatial factorization. They show how any of the existing algorithms for column segmentation can also be used for the row segmentation.

The non-rigid case is also included in this Section, since it can be considered as a multiple objects case. Techniques such as the Tomasi and Kanade factorization [110] can not be used in the non-rigid case, since they are all based on the rigidity assumption. As pointed out in [81], most of the proposed techniques for the non-rigid case are based on the fact that any configuration of the shape can be explained as a linear combination of basis shapes that define the principal modes of deformation of the object.

In [20], Bregler et al. present an approach to recover non-rigid shape from a single view video sequence. The key point is that it does not require multiple views nor a priori model as in previous approaches that also aim to obtain the non-rigid 3D shape. They tackle the 3D shapes recovering from 2D images and the parametrization of non-rigid shape deformation as a unique problem. The 3D non-rigid shape is recovered under scaled orthographic projection, using the fact that the 3D shape in each frame is a linear combination of a set of K basis shapes. Under this model, the 2D tracking matrix is of rank $3K$ and can be factored into 3D pose, object configuration and 3D basis shapes using the SVD under several factorization steps.

Brand [13] proposes an algorithm for tracking and 3D reconstructing non-rigid surfaces that have very little texture. The idea is to make equivalent the problem of non-rigid 3D structure from motion and the 2D optical flow estimation, yielding a combined non-rigid structure-from-intensities problem which can be solved via matrix decompositions. Thus, a factorization algorithm for 3D non-rigid structure and motion from video that finds 2D correspondences by enforcing 3D geometric constraints is presented.

In [111], Torresani et al. exploit low-rank constraints for low-level tracking, prediction of missing low-level features and non-rigid 3D reconstruction. The authors introduce three techniques able to reconstruct 3D deformable steps and their motion from 2D measurements. One drawback of the technique is that the number of basis shapes needed K is not known a priori. In their experiments, the value of K is increased until the error is below a given threshold.

Del Bue and Agapito [29] extend existing non-rigid factorization algorithms to the stereo camera formulation and present an algorithm to decompose the matrix of trajectories into the 3D shape and the motion corresponding to the left and right cameras. They use the fact that both cameras view the same structure and that the relative orientation between them is fixed. They show that the results are improved with the stereo approach.

In [124], Xiao et al. present a closed form solution to the problem of non-rigid shape and motion recovery form a single camera video. In practice, the non-rigid objects deform with certain shape basis. Many previous work (e.g., [13], [20]) use only the rotation constraints (that is the orthonormality constraints). In this paper, the basis constraints, which uniquely determine the shape bases, are proposed. Xiao et al. demonstrate that, under the weak-perspective projection model, a closed-form solution can be obtained by enforcing both the basis and the rotation constraints.

Brand [16] proposes a direct method to tackle the non-rigid structure from motion problem. The author points out that other proposals, such as [124] breaks down when data are noisy or the value of the rank of the basis is not correct. The approach proposed by Brand directly minimizes deviation from the required orthogonal structure of the projection/articulation matrix.

Lladó et al. [81] propose an approach for the 3D Euclidean reconstruction of deformable objects, under a perspective camera model with variable intrinsic parameters. They use the fact that some of the points on the object behave as rigid ones. This rigid points are used to obtain an initial estimate of the camera's varying internal parameters and the overall rigid motion. Furthermore, with the obtained initialization, the non-rigid structure can be upgraded to the metric space.

Kim and Hong [78] develop an approach for estimating the approximate average shape and motion of a deforming object, solving thus the ambiguity problem in non-rigid recovery. The authors introduce a novel concept, called *Degree of Nonrigidity* (DoN). Based on it, an iterative reweighted factorization method is proposed.

In a recent approach, Del Bue et al. [30] evaluate an algorithm able to segment rigid motion form non-rigid 2D trajectories obtained with a perspective camera. The RANSAC algorithm [36] is used to estimate the fundamental matrices of each pair of frames in the sequence and to segment the scene into rigid and non-rigid points. It is used the notion of DoN, introduced by Kim and Hong [78], to define a score for each trajectory. Concretely, this score is used to build a prior distribution of the conditional probability of each point in the object to be rigid or non-rigid. The authors point out that weak perspective effects and high proportion of non-rigid points can give a high misclassification ratio.

Tresadern and Reid [112] present a factorization method for articulated motion. They show how to detect articulated motion and how the rank of the matrix of trajectories gives the type of joint between the objects. Using the fact that the objects are connected, it is assumed that their relative depth is small compared to their distance from the camera. Hence, the affine projection can be used. The rank of the matrix of trajectories is approximated by finding a sharp drop in the singular values and their ratios. A method for self-calibration of the structure and motion to a metric space, which only uses linear constraints, is also proposed. However, that is only a sub-optimal solution that should be refined with a non-linear optimization technique, as point out the authors.

A different factorization approach to recovery articulated motion is proposed in [125]. Yan and Pollefeys [125] mark out that the global motion subspace of an articulated object is a combination of a number of intersecting rigid motion subspaces of the parts. Therefore, the goal is to cluster the motion subspaces and apply any robust reconstruction method of a rigid object to each segmented subspace. They use the Generalized Principal Component Analysis (GPCA) ([118], [119]) to cluster the subspaces. Thus, each group of trajectories belongs to an articulated part. Finally, the subspaces of each group of trajectories are intersected and the rank of the intersection allows to know whether the link between two parts is an axis or a joint (in case they are linked). In order to estimate the rank, a *model selection* inspired by the one proposed in [118] is used.

In a recent paper, Wang and Wu [120] propose to add a deformation weight constraint to the problem of 3D reconstruction of non-rigid objects. Furthermore, they present a constrained PowerFactorization [54] to recover the deformation structure in affine spaces. This technique allows to work with missing data. The main idea is to first reconstruct the object in an affine space. Then, the deformation parts are detected and separated from the rigid ones and the solution is upgraded to the Euclidean space.

Other techniques which are not purely factorization methods have been proposed to obtain the motion segmentation of feature points. In [39], Gear presents an algorithm for feature segmentation. It consists in computing the reduced row echelon form of the matrix of trajectories in order to easily identify the motion subspaces and the column vectors that lie in them. At the same time, the rank of the matrix can also be estimated. Gear considers more important to segment correctly the points than to estimate the rank exactly. The reduced row echelon form can be found by applying Gauss-Jordan elimination to the rows of the matrix of trajectories with partial pivoting. The main drawback of the algorithm is that all possible sorting columns should be tested in order to find the best solution, giving a combinatorially large task. Besides, results get worse if there are dependence between motions. However, for noisy scenarios this algorithm can not guarantee correct segmentation, as pointed out in [46].

Yan and Pollefeys [126] present an approach for motion segmentation that can deal with different kinds of motions: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. It consists in estimating a local subspace for each feature trajectory, and then compute an affinity matrix based on principal angles [43] between each pair of these estimated subspaces. Finally, the segmentation of the feature trajectories is obtained by applying spectral clustering [122] to this affinity matrix. Concretely, the *normalized cut criterion* defined in [103] is used. Although the presented results are promising and the approach can be used with different kinds of motions, there is one important fact that is not clear in the paper; how the recursive 2-way clustering is implemented. It seems that a prior knowledge about the number of objects in the scene is used in the algorithm. One of the drawbacks of this approach is that many parameters have to be set. For instance, the dimension of the projected space (in other words, the rank of the matrix of trajectories) and the dimension of the local subspaces and the number of neighbours that generate them. The authors propose to estimate the rank of the matrix of trajectories and the dimension of the local subspaces by using the *model selection* [71], but it is difficult to set the parameters to be valid for any sequence.

In a recent paper, Tron and Vidal [115] present a benchmark for the comparison of 3D motion segmentation algorithms. Concretely, they compare the performance of the GPCA [117] (presented in next Section), a RANSAC [36] based approach, in which the model to be fit is a subspace of a certain dimension, and the approaches proposed in [126] and in [107]. Since only the GPCA based approach can deal with missing data, the experiments are done with full data. The authors conclude that the best performing algorithm is the one proposed in [126], when the dimension of the projected subspace is 4 times the number of different motions. However, as in the previous approach [126], the number of different motions is used before estimating it.

Goh and Vidal [41] present an algorithm for segmenting motion of different types. The point trajectories are obtained for both affine and perspective views. They use nonlinear dimensionality reduction (NLDR) to obtain a low-dimensional representation of the union of all manifolds (under the perspective camera model, the trajectories associated with each motion live on a manifold instead of in a linear subspace, as in the affine camera model). In particular, an extension of the locally linear embedding algorithm (LLE) [96] is proposed: the Locally Linear Manifold Clustering Algorithm (LLMC). The LLE uses the fact that the local neighborhood of a point on the manifold can be well approximated by the affine subspace spanned by its k nearest neighbors. A comparison of the performances of the LLMC against other motion segmentation algorithms is provided, showing that the LLMC matches the performance of them.

In [79], Li et al. introduce an algorithm for segmentation of multiple rigid-body motions using multiple perspective views simultaneously. Their algorithm is an extension of the iterative algorithm presented in [113] to multiple motions. The idea is to alternate between the estimation of the depths and the segmentation of the point trajectories. The original algorithm [113] alternates between the estimation of motion and structure and the estimation of the depths. A comparative study of the performance of the proposed algorithm [79] with the techniques presented in [117], [126] and [107] is given. The proposed iterative algorithm outperforms the one presented in [107]. Compared to the other two techniques ([117] and [126]), the improvement with the proposed algorithm is not significant, as pointed out by the authors, as affine methods already give good results.

2.2.4 Multiple Objects, Missing Data Case

The multiple objects missing data case has not been as studied as the single object missing data case. This Section introduces proposed approaches that consider missing data.

Brand [14] proposes an incremental singular value decomposition that can deals with missing data. It can be used to develop a robust tracker based on the estimation of the subspace optical flow and also to handle missing data in non-rigid structure from motion factorization. The technique is presented in more detail in [17]. It consists in a novel approach for updating a rank-r SVD of a matrix by adding rows and/or columns of data. These new rows/columns may have missing entries and/or noise. The idea is to compute the SVD as data are added. The criterion to fill in missing entries consists of minimizing the rank value. An important fact that is the size of the data matrix is not known a priori. In a more recent paper [15], Brand proposes a very similar method and demonstrate its applicability to recommender systems. Instead of computing the rank-r SVD of a large matrix, he develops an exact rank-1 update which provides a linear-time construction of the whole SVD. The approach begins by sorting out the rows and columns of the data matrix so that a high density of data is accumulated in one corner. Then this initial full submatrix grows out of this corner by sequential updating with partial rows and columns. An imputation update that maximizes the probability of correct generalization is used to fill in the missing entries. One disadvantage of this technique is that the result depends on how the data is sorted, as pointed out in [54].

A hierarchical factorization method for recovering articulated hand motion under weak perspective projection is introduced by Zhou and Huang in [134]. In this method, each part of the articulated object is treated as an independently moving object. In a first step, Zhou and Huang propose to use any already presented technique in order to fill in the missing data, in case they exist, and to segment the trajectories into different objects. In a second step, the proposed hierarchical method aims at guaranteeing that the end of consecutive objects are linked in the recovered motion. Costeira and Kanade approach [26] is used to obtain the rotational component, but kinematic hand model constraints are needed to be integrated in order to obtain properly the translational component. One important requirement is that at least 4 non-coplanar feature points must be reliably tracked in order to recover the global pose and to initiate the hierarchical algorithm.

Gruber and Weiss [46] present an algorithm for multibody factorization with uncertainty and missing data which uses the expectation-maximization (EM) technique. As in the single object case formulation [45], prior knowledge can be easily added, improving the results even on challenging sequences. They show how the EM framework allows to place priors on both structure and motion and to deal with directional uncertainty and missing data.

In [117], Vidal and Hartley propose an algorithm for motion segmentation with missing data which uses PowerFactorization method [54] and the generalized principal component analysis (GPCA)([118], [119]). In a first step, the trajectories are projected onto a five dimensional subspace by using the PowerFactorization. This is justified by the fact that the trajectories of a rigid moving object generate a subspace of dimension 4 at most. Then, projecting onto a generic five-dimensional subspace preserves the clustering of the motion subspaces. Once the missing data are filled in, the GPCA is used to obtain the motion segmentation of trajectories. The algorithm can handle degenerate and dependent motions. In [118], an extension of the algorithm to low-dimensional subspaces is presented. One of the main drawbacks of the GPCA, as pointed out in [126] and [115], is that its performance gets worse as the number of objects in the scene increases.

In a recent paper [47], Gruber and Weiss present an extension of the aforementioned approach [46] that consists of incorporating non-motion cues into 3D motion segmentation. Concretely, they propose a constrained factorization with priors that uses the EM algorithm. The idea is to define an energy function that consists of two parts: the first part is due to 3D motion and the second part is due to spatial coherence. In this second part, 2D affinities between pixels, similar to the ones presented in [103], are defined. The goal is to minimize the energy function. Although in the case of 2 objects, it is guaranteed to find a local maximum, this is not guarantee in the case of more objects. One important advantage of this approach is that it can deal properly with noisy scenarios and with missing data. Furthermore, no assumptions are made regarding the rank of the motion matrix. A comparison to the previous approach [117] shows that spatial priors addition improve the results obtained in the motion segmentation.

Del Bue [28] presents a method to estimate average 3D shapes from non-rigid motion in the case of missing data. Firstly, an initial estimate of the 3D shape and motion of the object is obtained with a power method (based on the PowerFactorization technique). Secondly, this initial estimation is refined by using non-linear optimization. As far as our knowledge, Del Bue is the first in tackling the shape from non-rigid motion with missing data.

Tables 2.1, 2.2 and 2.3, summarize the main features of the studied factorization techniques focused on the SFM problem.

Table 2.1: summary of different methods of factorization I.

Method	camera model	object	number objects	missing data	noisy data	used tools and features
[110]	orthographic	rigid	single	yes	no	SVD
[91]	parapersp. persp. iterative	rigid	single	no	no	SVD
[106]	perspective	rigid	single	no	no	SVD, fundamental matrices epipoles
[113]	perspective	rigid	single	no	no	SVD, 3D lines
[57]	perspective	rigid	single	no	no	iterative
[25]	perspective	rigid	single	no	no	affine iterations
[93]	affine	rigid	single	no	no	lines
[3]	orthographic	rigid	single	no	no	surface-based
[87]	affine	rigid	single	no	yes	point, lines planes
[70]	affine	rigid	single	yes	no	point, lines conics
[6]	affine	rigid	single	no	yes	SVD
[1]	perspective	rigid	single	yes	yes	IRLS
[2]	affine	rigid (piecewise planar)	single	no	yes	rank 1 weighted factorization
[4]	affine	rigid	single	no	yes	power method
[45]	affine	rigid	single	yes	yes	temporal prior, EM
[61]	affine	rigid	single	no	yes	outliers detection
[62]	affine	rigid	single	no	yes	reprojec. error
[60]	affine	rigid	single	yes	yes	outliers correction
[86]	affine	rigid	single	no	no	sequential, SVD
[80]	affine	rigid	single	no	no	sequential, SVD and RLS
[66]	affine	rigid	single	yes	no	r-tuple
[24]	affine	rigid	single	yes	yes	inputation
[67]	affine	rigid	single	yes	yes	DP criterion
[89]	affine	rigid	single	yes	no	Wiberg
[50]	affine	rigid	single	yes	no	Row-column
[54]	affine	rigid	single	yes	no	PF
[23]	affine	rigid	single	yes	no	Damped Newton hybrid

Table 2.2: summary of different methods of factorization II.

Method	camera model	object	number objects	missing data	noisy data	used tools and features
[44]	affine	rigid	single	yes	no	global approach
[83]	perspective	rigid	single	yes	no	two step combining [66] and [106]
[108]	affine	rigid	single	yes	yes	geometric correlations 2D-3D shape
[18]	affine	rigid	single	yes	no	EM
[84]	affine, perspective	rigid	single	yes	no	rank 4 submatrices
[51]	affine	rigid	single	yes	no	F_A-closure
[12]	affine	rigid	multiple	no	yes	SVD
[26]	affine	rigid	multiple	no	no	SVD,interaction matrix
[53]	affine	rigid	multiple	no	no	SVD, linear constant speed
[64]	affine	rigid	multiple	no	no	SVD, QR
[63]	affine	rigid	multiple	no	no	SVD, QR
[72]	affine	rigid	multiple	no	yes	geometric AIC
[74]	affine	rigid	multiple	no	no	AIC
[73]	affine	rigid	multiple	no	yes	AIC
[130]	affine	rigid	multiple	no	yes	degeneracies
[107]	affine	rigid	multiple	no	yes	degeneracies
[132]	affine	rigid	multiple	no	yes	trajectories horitzontally stacked
[131]	affine	rigid	multiple	no	yes	temporal consistency
[20]	scaled orthographic	non-rigid	—	no	no	basis shapes
[13]	affine	non-rigid	—	no	no	morphable
[111]	affine	non-rigid	—	no	no	tracking
[29]	affine	non-rigid	—	no	no	stereo
[124]	weak-perspective	non-rigid	—	no	no	basis constraints set of rigid
[16]	affine	non-rigid	—	no	yes	tracking 3D surfaces
[81]	perspective	non-rigid	—	no	no	rigid points as initialization

Table 2.3: summary of different methods of factorization III.

Method	camera model	object	number objects	missing data	noisy data	used tools and features
[78]	affine	non-rigid	—	no	no	Degree of Nonrigidity
[30]	perspective	non-rigid	—	no	yes	RANSAC, F matrix
[112]	affine	art.	—	no	no	SVD
[125]	affine	art.	—	no	no	SVD, GPCA
[120]	affine	non-rigid	—	yes	no	constrained PF
[39]	affine	rigid	multiple	no	no	row echeleton
[126]	affine	all	—	no	no	local subspace, $Ncuts$
[115]	affine	all	multiple	yes	no	comparison
[41]	affine, perspective	all	multiple	yes	no	LLMC, NLDR
						iterative
[79]	perspective	rigid	multiple	no	no	extensions of [106]
[14]	affine	non-rigid	—	yes	yes	incremental SVD
[134]	weak-persp.	art.	—	yes	no	hierarchical
[46]	affine	rigid	multiple	yes	yes	EM
[117]	affine	rigid	multiple	yes	no	PF, GPCA
[47]	affine	rigid	multiple	yes	yes	2D nonmotion cues
[28]	affine	non-rigid	—	yes	yes	PF

Chapter 3

Alternation Technique

This Chapter presents in detail the Alternation technique, which is widely used along the thesis to obtain the best low-rank approximation of a given missing data matrix. This low-rank matrix approximation can be used to fill in missing entries in the data matrix. However, in some cases, the main objective is to factorize the missing data matrix in two factors, without considering the obtained filled in entries with their product.

First of all, Wiberg's algorithm [123] is introduced, since it can be considered the origin of Alternation. Then, the Alternation algorithm is presented, both in matrix and rows formulations. Additionally, different missing data distributions are studied, together with their influence to the performance of the technique. Finally, the denoising capacity of the Alternation is empirically shown.

3.1 Wiberg's Algorithm

Wiberg [123] introduces an iterative algorithm for the computation of principal components of a data matrix with missing observations. It is based on Ruhe and Wedin's approach [97]. The key point of the algorithm is that variables are separated into two sets. At each iteration, in a first step, the optimization with respect to one set of variables is performed. Then, in a second step, the correction of the second set of variables is carried out. The main ideas of the Wiberg's algorithm are presented below following its formulation.

Given a data matrix Y, which consists of observations of n individuals from a p-variate normal distribution with mean vector $\bar{\mu}$ and covariance matrix Σ, the r-component model is:

$$Y = e\mu^t + \tilde{U}_{n\times r}\tilde{S}_{r\times r}\tilde{V}^t_{p\times r} \tag{3.1}$$

where $\tilde{S} = diag(\sigma_i)$ is an $r \times r$ diagonal matrix containing the singular values σ_i of $Y - e\mu^t$, being $\sigma_i \geq \sigma_{i+1}$, μ is the maximum likelihood approximation to the mean vector $\bar{\mu}$ and e is a vector with all ones. The aim is to determine $\tilde{U}$, $\tilde{S}$ and $\tilde{V}$ such that minimize the following expression:

$$\|Y - e\mu^t - \tilde{U}\tilde{S}\tilde{V}^t\|_E \tag{3.2}$$

where $\|\cdot\|_E$ is the Euclidean matrix norm [43].

The singular value decomposition (SVD) of the centered data matrix $Y - e\mu^t$ gives the solution to this problem when there are no missing data in Y. Unfortunately, this method can not be used when Y has missing data. The expression to minimize in that case is:

$$\varphi(\mu, U, V) = \frac{1}{2}\sum_{j,k}(Y_{jk} - \mu_k - u_j^t v_k)^2 \tag{3.3}$$

where j and k correspond to the known entries $Y(j,k)$, and v_k and u_j are defined as:

$$U = \tilde{U}\tilde{S}^{1/2} = \begin{bmatrix} u_1^t \\ \vdots \\ u_n^t \end{bmatrix}, \quad V = \tilde{V}\tilde{S}^{1/2} = \begin{bmatrix} v_1^t \\ \vdots \\ v_p^t \end{bmatrix} \tag{3.4}$$

Since (3.3) defines a least squares problem, any algorithm for nonlinear least squares can be applied. The classical way to obtain a numerical solution of a nonlinear least squares problem is the Gauss-Newton algorithm.

Rewriting the matrices from above as vectors, the minimization problem (3.3) can be written as

$$\varphi = \frac{1}{2}f^t f \tag{3.5}$$

where

$$f = y - \hat{\mu} - Fu = y - G\tilde{v} \tag{3.6}$$

being

$$u = \begin{bmatrix} u_1^t \\ \vdots \\ u_n^t \end{bmatrix}, \quad \tilde{v} = \begin{bmatrix} \tilde{v}_1^t \\ \vdots \\ \tilde{v}_p^t \end{bmatrix}, \tag{3.7}$$

$$\tilde{v}_k = \begin{bmatrix} v_k & \mu_k \end{bmatrix}, \quad \hat{\mu}_i = \mu_{k(i)}, \tag{3.8}$$

$$F = \frac{\partial f}{\partial u}, \quad G = \frac{\partial f}{\partial v} \tag{3.9}$$

Since the goal is to minimize (3.5), its partial derivatives are computed:

$$\dot{\varphi} = \begin{bmatrix} \frac{\partial \varphi}{\partial u} \\ \frac{\partial \varphi}{\partial v} \end{bmatrix} = \begin{bmatrix} F^t(Fu - (y - \hat{\mu})) \\ G^t(G\tilde{v} - y) \end{bmatrix} \tag{3.10}$$

The Gauss-Newton algorithm replaces the nonlinear problem (3.3) by a sequence of linear ones, taking advantage of the sparse, structured derivatives of the devised object function φ (3.5). Hence, the variables are separated into two sets and the following algorithm is obtained:

Wiberg's algorithm: Given $Y_{n \times p}$ a data matrix that contains observations of n individuals from a p-variate normal distribution:

1. Initialize $\tilde{v}$.
2. Compute u: $u = F^{+}(y - \hat{\mu})$, where F^{+} is the *pseudo-inverse* of F [43].
3. Stop if converge.

   Otherwise, compute dv such as minimizes $\|Q_F G dv - Q_F y\|_2$, where Q_F is the projector to the column space of F. That is: $Q_F = I - F(F^t F)^{-1} F^t$.

   Update $\tilde{v}$: $\tilde{v} = \tilde{v} + dv$,
4. Repeat the steps 2-3 until the convergence is achieved.

Solution: u and v minimize $\varphi(\mu, U, V) = \frac{1}{2} \sum_{j,k} (Y_{jk} - \mu_k - u_j^t v_k)^2$

More details of the original approach can be found in [98].

Shum et al. [104] present an outline of Wiberg's algorithm and develop a method for 3D object modeling based on it. This is the first work on the computer vision framework based on Wiberg's algorithm. They show that object modeling from a sequence of range images can be thought as a problem of principal component analysis with missing data (PCAMD). Wiberg's formulation is modified and generalized to a weighted least squares (WLS) minimization which simultaneously recovers object shape and transformation among different views.

In a recent paper, Okatani and Deguchi [89] present in detail Wiberg's algorithm focused on the matrix factorization problem. Their experiments show the good performance of this algorithm compared to the Levenberg-Marquardt (LM) technique. They propose to use it as a standard algorithm for some computer vision problems. Furthermore, authors point out the misunderstanding in the computer vision community of associating some techniques to Wiberg's approach. In order to show that, they expose in [89] some differences between the Alternation technique, which they call *Alternated Least Squares* (ALS) technique, and the Wiberg's algorithm.

However, although the objectives and methods to solve Wiberg's and Alternation algorithms are different, both techniques are quite related. In some way, Wiberg's algorithm can be thought as Alternation origin, as considered in previous papers. For instance, Guerreiro and Aguiar [50] mention that their algorithm is similar to Wiberg's one [123] and Buchanan and Fitzgibbon [23] cite Wiberg [123] as the origin of Alternation technique. As far as we are concerned, Buchanan and Fitzgibbon are the first in calling this technique Alternation.

3.2 Alternation Technique's Algorithm

This Section presents the algorithm and properties of the Alternation technique. Given a matrix $W_{m \times n}$ with rank $r < m, n$, the aim is to find $A_{m \times r}$ and $B_{r \times n}$ such that minimize the following expression:

$$\|W - AB\|_F^2 \tag{3.11}$$

Working with missing data, the expression to minimize is:

$$\|W - AB\|_F^2 = \sum_{i,j} |W_{ij} - (AB)_{ij}|^2 \tag{3.12}$$

where i and j correspond to the index pairs where W_{ij} is defined.

The SVD gives the solution to this problem when the matrix W is complete [110]. The Alternation technique can also be used to solve the above problem, even if there are missing data in W. This technique uses the fact that if one of the factors A or B is known, the other factor can be computed by solving the least squares problem (3.12).

There is a very interesting relationship between the Alternation and the *orthogonal power iteration* [43], as it is shown in [50] and [54]. This relationship is exposed below in order to justify that the Alternation gives the closest low-rank approximation to W, in the least-squares sense.

The *Power Method* [43] is a well known technique for finding the dominant eigenvector of a matrix X. Starting from a random vector u_0, the *Power Method* consists in repeatedly applying X to u and normalizing the result:

$$u_{k+1} = Xu_k / \|Xu_k\| \tag{3.13}$$

The *orthogonal power iteration* [43] consists in extending the *Power Method* to find the r-dimensional dominant subspace of a given matrix. Applying it to a symmetric positive semi-definite matrix WW^t, it starts with a random matrix U_0 with r orthonormal columns and iteratively multiply by WW^t and re-orthonormalize the columns:

$$U_k = (WW^t)U_{k-1}N_k \tag{3.14}$$

where N_k is an upper triangular matrix that makes the columns of U_k orthonormal. Actually, N_k is a normalizing matrix that is found by the Gram-Schmidt process [43], which is equivalent to *QR decomposition* [43] of $(WW^t)U_{k-1}$.

Using the above iteration, it can be seen that, to find the dominant space of WW^t is equivalent to the two-step algorithm presented below.

Alternation algorithm: Given a matrix $W_{m \times n}$ of rank $r < m, n$, the algorithm starts with an initial random $m \times r$ matrix A_0 (analogously with a random B_0) and repeats the steps 1-2 until the product A_kB_k converges to $W_{m \times n}$:

1. Compute the matrix B_k:

$$B_k = (A_{k-1}^t A_{k-1})^{-1} (A_{k-1}^t W) \tag{3.15}$$

2. Compute the matrix A_k:

$$A_k = W B_k^t (B_k B_k^t)^{-1} \tag{3.16}$$

3. Stop if the product $A_k B_k$ converges to W.

   Otherwise, set $k = k + 1$ and go to step 1.

Solution: The product $A_k B_k$ is the closest rank-r matrix to W, in the least-squares sense.

If the expression (3.15) is substituted to (3.16), the following single step is obtained:

$$A_k = W W^t A_{k-1} (A_{k-1}^t W W^t A_{k-1})^{-1} A_{k-1}^t A_{k-1} \tag{3.17}$$

As pointed out Guerreiro and Aguiar [50], the above expression shows that the two step algorithm is equivalent to apply the *Power Method* [43] to the matrix WW^t, where the factor $(A_{k-1}^t W W^t A_{k-1})^{-1} A_{k-1}^t A_{k-1}$ corresponds to the normalization N_k of equation (3.14). Hence, the Alternation can be seen as a generalization of the *Power Method*.

Notice that, although the computation of the factors is different to that of Wiberg's algorithm (here the Gauss-Newton method is not used), the variables (matrix factors) are also computed separately, in two steps. The key point is that each B_k and A_k is the least-squares solution to a set of equations of the form $W = AB$ and, at each step, the expression (3.12) is minimized. One of the advantages of Alternation is that it has a very rapid convergence if the rank of the initial matrix W is close to r, as showed in [54] (only one iteration is needed in the full data case). Actually, the fact of solving for each factor A (or B), while holding the other factor fixed, provides a minimum in each direction in a single step. Moreover, since at each iteration the algorithm consists of only matrix multiplication and inversion of small ($r \times r$) matrices, it is a fast algorithm. Of course, if r is large, the algorithm becomes slower.

The Alternation technique allows to obtain the two factors A and B when the SVD can not be applied. At the same time, Alternation can also be used as an imputation method, since missing entries can be recovered with the product of the obtained factors A and B. For instance, in [50] the Alternation is used to find a low-rank matrix approximation to W and the missing entries are filled with the product: AB. This is also used in Chapter 5 and in Chapter 6, as shown later.

Unfortunately, the decomposition obtained with the Alternation is not unique, as in the SVD case, since any $r \times r$ invertible matrix also gives a valid decomposition:

$$W = AB = \widehat{A}QQ^{-1}\widehat{B} \tag{3.18}$$

The Q matrix can be computed by imposing constraints on A or B, depending on the nature of the factors. In the particular adaptation of Alternation to the SFM problem, which is studied in Chapter 4, Q is computed with a *normalization* step similar to the one used in the original formulation [110] (see equation (2.19), in Section 2.1.1).

The main advantage of working with this two-step algorithm is that it can be extended to missing data by observing a link with least-squares problems, which is not obvious with the *orthogonal iteration* method. Furthermore, the updates of B given A (analogously A given B) can be done by solving a least squares problem for each row of B independently. Therefore, the missing entries in W correspond to omitted equations. The Alternation algorithm considering the rows of A and columns of B independently, instead of taking the whole matrices, is presented below.

Alternation algorithm, row-column formulation: Given a matrix $W_{m\times n}$ with rank $r < m, n$, the algorithms starts with a random matrix of dimensions $m \times r$, A_0 (analogously with a random B_0). Set k =1 and repeat the following steps until the product A_kB_k converges to $W_{m\times n}$:

1. Compute the rows of B_k independently:

   To compute the j-column of B_k, referred to as $\mathbf{b}^j$ hereinafter, the known entries in the j-column of W are used. Concretely, for each i such as $W(i, j)$ is known, an equation is formulated, using the corresponding rows of the known matrix A_{k-1}.

   The following expression gives the solution for the j-column of B_k:

   $$\mathbf{b}^j = (A_{k-1}^t A_{k-1})^{-1}(A_{k-1}^t \mathbf{w}^j) \tag{3.19}$$

   where $\mathbf{w}^j$ is the j-column of W. It should be remarked that only the known entries in this column and the corresponding rows in A_{k-1} are considered.

2. Given B_k, compute A_k analogously:

   To compute the i-row of A_k, $\mathbf{a}_i$ hereinafter, each of the known entries in the i-row of W is taken to formulate an equation using the known factor B_k. The i-row of A_k is computed as:

   $$\mathbf{a}_i = (\mathbf{w}_i B_k^t)(B_k B_k^t)^{-1} \tag{3.20}$$

   where $\mathbf{w}_i$ is the i-row of W and only its known entries and the corresponding columns in B_k are used.

3. Stop if the product A_kB_k converges to W.

   Otherwise, set $k = k + 1$ and go to step 1.

Solution: The product A_kB_k is the closest rank-r matrix to W, in the least-squares sense.

Notice that the number of equations used to compute each row and column of each factor A and B, depends on the number of known entries in each row/column of W. Actually, notice also that the number of unknowns when computing each row $\mathbf{a}_i$ and each column $\mathbf{b}^j$ is r, which is the length of the computed row or column. Hence, if there is only a few missing entries in the corresponding row or column of W, a global minimum is achieved and the factors are properly computed. However, with a high percentage of missing data, only a few equations can be formulated and the method may converge to a local minimum. Although in theory only r equations are needed, in practice degenerate cases, such as linearly dependent equations, can occur and more than r equations may be required.

It should be highlighted that not only the percentage of missing data affects the performance of the Alternation, but also, and more remarkably, the distribution of the missing data. In general, when the missing data is *randomly* distributed, the obtained results are good, while the percentage of missing data is not very high. On the contrary, if missing data are concentrated in some areas of the given matrix W (e.g., *banded* structure in the SFM problem, see Chapter 4) the Alternation technique could fail. A study of Alternation robustness to different missing data distributions is presented in next Section.

3.3 Missing Data Distribution Variants

Two different missing data distributions are considered in this Section. This is an important fact that should be taken into account, since the performance of the Alternation technique depends on missing data distribution, as mentioned above.

Several Alternation-based approaches proposed in the literature present experiments assuming a *random* distribution of the missing data. Fig. 3.1 (left) shows an example of this distribution. The Alternation technique gives better results when the missing data presents this distribution, which is found in some applications such as recommender systems or *DNA microarray* data (studied in Chapter 6).

Unfortunately, other problems tackled with Alternation do not present such a random distribution (e.g., SFM, see Chapter 4). An example of *banded* distribution can be seen in Fig. 3.1 (right). This distribution can give columns with less than r known elements and that will give undesirable results.

A study of the performance of the Alternation depending on the data distribution is provided. Since the presented Alternation is not focused on any application, a synthetic data matrix is considered for this experiment. Concretely, two matrices $A_{m\times r}$ and $B_{r\times n}$ are synthetically generated with points that belong to a normal

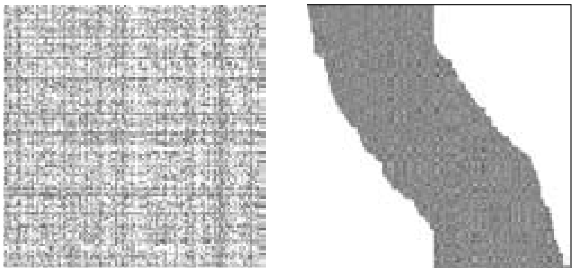


Figure 3.1: matrices with 50% of missing data; (left) *random* distribution; (right) *banded* distribution.

distribution. The initial data matrix W is defined by the product: AB.

The number of rows m and columns n are 200 and r is set to 5, in the current experiment. Taking the full matrix W, missing data matrices are generated considering both missing data structures, in order to compare the performance of Alternation in each case. In the case of *banded* structure data, given a matrix where all values are known, different percentages of missing data are generated by automatically removing parts of random columns. The removing process randomly selects a cell in the given column, dividing it into two parts. One of these parts is randomly removed—in the SFM problem, this removing process simulates features missed by the tracker or new features detected after the first frame, respectively.

Concretely, 50 attempts are repeated and the *root mean square error* (*rms*) at each attempt is computed. It is defined as follows:

$$rms = \frac{\|W - AB\|_F}{\sqrt{p}} = \sqrt{\frac{\sum_{i,j} |(W)_{ij} - (AB)_{ij}|^2}{p}} \tag{3.21}$$

where i and j correspond to the index pairs where W_{ij} is defined and p is the number of those pairs in W.

The results obtained considering different percentages of missing data are plotted in Fig. 3.2. Concretely, polygonal figures enclose data in between lower and upper quartiles (medians are represented by horizontal lines in thinner regions). It can be seen that with low percentages of missing data, the results are similar in both cases. When the percentage of missing data is equal or higher than 40%, the performance with the *banded* missing data structure becomes worse. Notice that with a percentage of missing data of 90%, the *rms* values are similar in both cases. This is due to the fact that, with such a high percentage of missing data, a similar matrix W is obtained with both distributions.

This is a synthetic, ideal experiment. Unfortunately, in real situations, the data matrix can present more extreme *banded* structure. Notice that in this generated *banded* matrix, most columns have many known values. Furthermore, data is quite symmetric in this example. When the data matrix corresponds to a real data, even more extreme distribution should be faced out (e.g., a column with less than r ele-

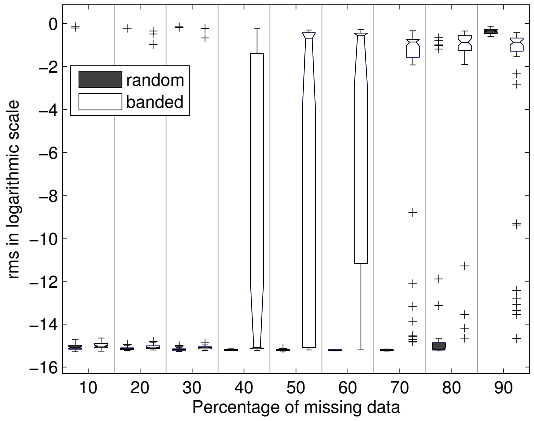


Figure 3.2: *rms* in logarithmic scale considering different percentages of missing data and for the random and *banded* missing data distribution.

ments).

3.4 Denoising Capability of the Alternation

This Section presents a study about the *filtering* (also referred to as *denoising*) capacity of the Alternation technique, which has already been explored in [60].

Chen and Suter [24] present a study about the *denoising* capacity of the SVD versus missing data. Concretely, they point out that when the SFM is tackled through factorization of a complete matrix, all the feature points are treated uniformly. Thus, most of the noise can be suppressed if the size of the matrix of trajectories is large enough. However, the accuracy of this denoising capacity degrades as the percentage of missing data increases. Since the Alternation and the SVD give the same solution in the full data case, it seems natural that the Alternation has also a denoising capacity.

Experimental results are carried out in order to show the *filtering* capacity of the Alternation. Different levels of Gaussian noise—standard deviation σ with a value from $\frac{1}{2}$ to 10 and zero mean—are added to the entries of W. The obtained noisy matrices are denoted as $\widehat{W}$. The Alternation technique is applied to each of these matrices and the root mean square error *rms* is computed:

$$rms = \frac{\|\widehat{W} - AB\|_F}{\sqrt{p}} \tag{3.22}$$

where p is the number of elements in $\widehat{W}$.

Furthermore, the root mean square error is computed considering the free of noise matrix W instead of $\widehat{W}$. This measure allows to study if noisy entries have been

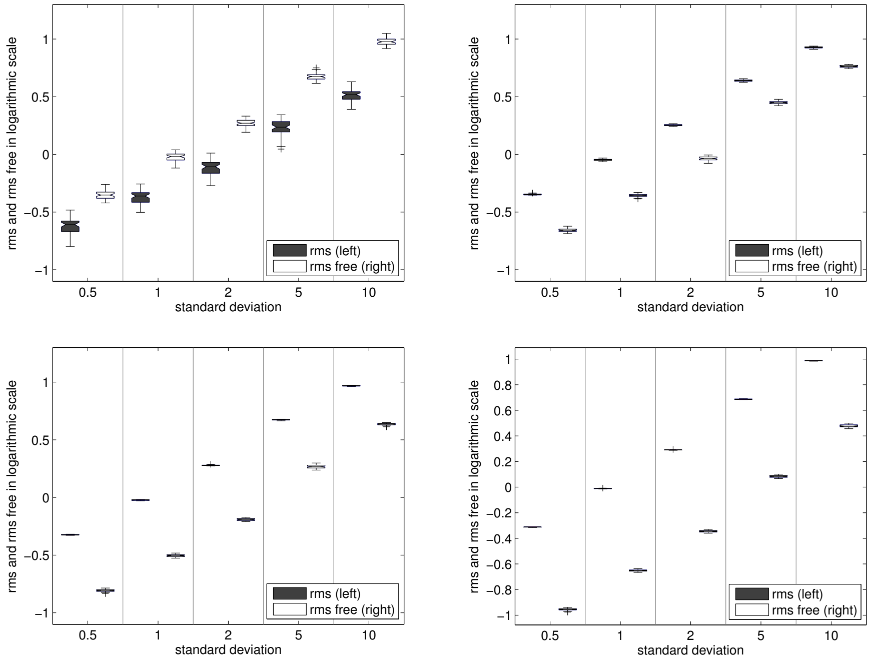


Figure 3.3: rms and rms_{free}, in logarithmic scale, considering different standard deviation values and fixing the size of the matrix; (top) size of W: 10×10 and 50×50; (bottom) size of W: 100×100 and 200×200.

corrected (filtered). This error is referred to as rms_{free} hereinafter and it is defined as follows:

$$rms_{free} = \frac{\|W - AB\|_F}{\sqrt{p}} \tag{3.23}$$

Notice that in case Alternation filters the noise, rms_{free} should be smaller than rms.

As before, the rank is set to 5 and for each setting, 50 attempts are repeated. Three different experiments are carried out, depending on the studied parameters in each case.

The first experiment consists in fixing the size of the data matrix, while varying the standard deviation of the gaussian noise added to W. Fig. 3.3 shows the obtained rms and rms_{free}. It can be appreciated that the Alternation does not filter noise in the case of a data matrix with a size 10×10 (Fig. 3.3 (top-left)). On the contrary, noisy data are filtered when the size of W is larger (for any tested standard deviation). Notice that the distance between rms and rms_{free} becomes smaller as the amount of noise increases. That is, it is more difficult to *denoise* matrices with a large amount of noise.

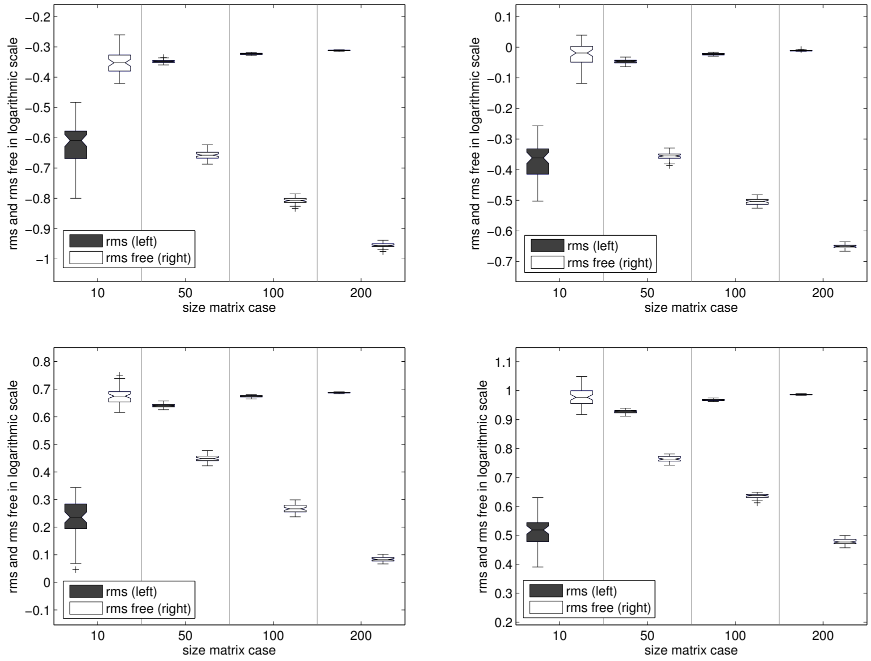


Figure 3.4: rms and rms_{free}, in logarithmic scale, considering different matrix sizes and fixing the standard deviation value; (top) $\sigma = 1/2$ and $\sigma = 1$; (bottom) $\sigma = 5$ and $\sigma = 10$.

A similar experiment consists in fixing the amount of noise added to W, while taking matrices with different sizes. The obtained results are plotted in Fig. 3.4. As in the previous experiment, the Alternation technique can not filter noisy data in the case of a very small matrix (concretely, a matrix with only 10 rows and columns), as it can be seen in Fig. 3.4. It should be remarked that, as the size of the matrix increases, the rms keeps nearly constant, while the rms_{free} becomes smaller. This is due to the fact that the noise can be more easily filtered when working with large matrices.

The above experiments take only full data matrices. An experiment that considers different percentages of missing data in the initial matrix is also provided. Fig. 3.5 shows the rms values obtained considering different percentages of randomly distributed missing data. In this experiment, the matrix has 100 rows and columns, the standard deviation is $\sigma = 1$ and the rms defined in equation (3.21) is used. It can be seen that, as expected, the filtering capability of the Alternation gets worse as the percentage of missing data increases (see increases in rms_{free}). Concretely, it filters the noise while the percentage of missing data is below 80%.

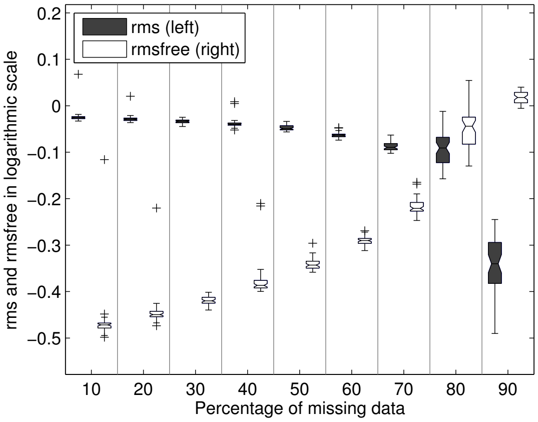


Figure 3.5: rms and rms_{free}, in logarithmic scale, considering different percentages of randomly distributed missing data. The standard deviation is $\sigma = 1$ and the size of the matrix W is set to 100×100.

3.5 Summary

This Chapter presents the algorithm and properties of Alternation technique. It should be remarked that the Alternation can be used to factorize a missing data matrix, but also to fill in its missing entries, with the product of the recovered factors. Furthermore, different missing data distributions are presented and a study of the performance of the Alternation with each distribution is provided. Concretely, random and non-random missing data distributions are studied. Experimental results show that a random distribution of the missing data gives better results. Finally, the *filtering* capacity of this technique is empirically shown.

Chapter 4

SFM in the Single Object Case with Missing Data

This Chapter tackles the Structure from Motion problem for the single object case through factorization, focusing on matrices of trajectories that contain a high percentage of missing data.

Several techniques have been proposed for tackling the Structure from Motion problem through factorization in the case of missing data. However, when the percentage of unknown data is high, most of them may not perform as well as expected. Focusing on this problem, an iterative multiresolution scheme, which aims at recovering missing entries in the original input matrix, is proposed. Information recovered following a coarse-to-fine strategy is used for filling in the missing entries. The objective is to improve the results by applying a factorization technique to the partially or totally filled matrix instead of to the original input one. An evaluation study about the robustness to missing and noisy data is reported. Experimental results obtained with synthetic and real video sequences are presented to show the viability of the proposed approach.

4.1 Introduction

The *Structure From Motion* (SFM) problem consists in extracting the 3D shape of a scene as well as the relative camera-object motion from trajectories of tracked features. In the computer vision context, factorization is a theoretically sound method addressing this problem. Since it was introduced by Tomasi and Kanade [110] many variants have been presented in the literature (e.g. [91] for the case of paraperspective camera model; a sequential factorization method in [86]; [26] and [53] for the multiple object case, etc). The 2D image coordinates of a set of 3D features (points in general, but also lines [93] and planes [87]) are stacked into a matrix of trajectories, where every row represents a frame of the sequence and every column a given feature. **Factorization techniques** aim at expressing this matrix of trajectories as the product of two unknown matrices, namely, the relative camera-object motion at each frame (M) and the 3D shape (S) of the object:

$$W_{2f \times p} = M_{2f \times r} S_{r \times p} \tag{4.1}$$

where f, p are the numbers of frames and feature points respectively and r the rank of W. Given an input matrix W, the goal is to find the factors M and S that minimize $\|W - MS\|_F^2$, where $\| \cdot \|_F$ is the Frobenius matrix norm [43]. These factors can be estimated by using the fact that W has a reduced rank and due to constraints derived from the orthonormality of the camera axes.

The *Singular Value Decomposition* (SVD) gives the closed-form solution to this problem when there are not missing entries. Unfortunately, trajectories are often incomplete or split due to objects occlusions, missing on the tracking or simply because they exit the camera field of view. Hence other methods need to be used in these cases. As presented in Section 2.2.2, several factorization techniques have been proposed to tackle the problem of missing data.

The main drawback of factorization techniques is found working with a large percentage of missing data; the obtained solutions get worse as the percentage of missing data increases. In this Chapter, an iterative multiresolution scheme is presented. It allows to fill in missing data in the matrix of trajectories. The key point of the proposed approach is to work with submatrices, instead of with the whole matrix of trajectories. That is, reduced sets of feature points along a few number of consecutive frames are selected. Then, for each set (submatrix), the 3D reconstruction and the camera motion corresponding to the used feature points are obtained by applying a factorization technique. Concretely, the Alternation technique introduced in Chapter 3 is adapted to the SFM problem. Missing entries in each selected set can be filled in just by multiplying the recovered shape and motion matrices. One of the main contributions over the two preliminary approaches ([68] and [69]) is that only submatrices with a percentage of missing data below 50% are used, in order to assure good recovered factors. Furthermore, a more extended evaluation study is reported. On the one hand, and only in the synthetic case, the recovered factors M and S are studied and compared to the ground truth ones. On the other hand, the goodness of the recovered entries is studied by taking into account both the initially known entries and also the initially missing ones. The latter is only possible when the full matrix is known in advance.

The proposed approach should be seen as a pre-processing technique; that is, firstly the matrix of trajectories is partially or totally filled in with the proposed iterative multiresolution scheme. Then, any factorization technique could be applied in order to obtain the structure and motion of the whole matrix. The final goal is to improve results when the factorization is applied to the matrix filled in with the proposed scheme, instead of applying it directly to the original input matrix, which has a higher percentage of missing data.

The remainder of the Chapter is organized as follows. Section 4.2 introduces the Alternation technique adapted to the SFM problem. This technique is used in the iterative multiresolution scheme to fill in missing data and also, at the final step, to factorize the whole matrix into the motion and shape matrices. The proposed iterative multiresolution scheme is presented in Section 4.3. An evaluation study of the performance of the proposed scheme is provided in Section 4.4, both for synthetic and real data. Finally, conclusions are summarized in Section 4.5.

4.2 Alternation Adapted to the SFM Problem

This Section presents the Alternation technique adapted to the SFM problem. This technique was introduced in Chapter 3, without focusing on any application. In this case, the A and B factors correspond to the motion M and shape S matrices, respectively. The Alternation technique is used in the iterative multiresolution scheme to fill in missing data and also, at the final step, to decompose the whole matrix into the motion and shape matrices. First of all, a brief summary of several variants of the Alternation focused on the SFM problem are reported. Then, the *Alternation with motion constraints*, which consists in imposing the orthonormality of the camera axes at each iteration of the algorithm, is presented. Furthermore, a variant that includes constraints for imposing the *smoothness camera motion* is also proposed. An example to illustrate the performance of this variant is provided.

4.2.1 Variants of Alternation Focused on SFM

Several variants of the Alternation technique have been presented in the literature. In [23], and more extended in [22], a report of most of these variants, including their algorithm, is given. Here, variants focused on the SFM problem are briefly exposed.

Hartley and Schaffalitzky [54] suggest adding a normalization step between the two factors updates at each iteration. This particular Alternation technique is denoted as PowerFactorization. Furthermore, focusing on the SFM problem, the authors propose to impose that the last row of S is equal to $\mathbf{1}$ (where $\mathbf{1}$ represents a vector of 1). In a different approach, Huynh et al. [60] propose to add an update of W at each iteration, once the factors have been computed. Their objective is to correct the outlier entries in W. However, they do not consider the missing data case. Buchanan and Fitzgibbon [23] propose the *Alternation/Damped Newton Hybrid*, which combines the Alternation strategy with the *Damped Newton* method. A different Alternation approach is presented by Aanaes and Fisker in [1]. Given an initial measurement matrix W_0, the algorithm consists in computing the factors M and S separately. Additionally, at each iteration, the measurement matrix is modified in order to approximate a perspective camera model.

In Table 4.1, a brief summary of the studied variants of the Alternation technique to tackle the SFM problem is given.

Table 4.1: variants of Alternation technique to tackle the SFM problem.

Method	Characteristics	Objective
Hartley and Schaffalitzky [54]	PowerFactorization (normalization step)	SFM
Huynh et al. [60]	update W	to correct outliers in W
Buchanan and Fitzgibbon [23]	Damped Newton hybrid	SFM
Aanaes and Fisker [1]	modify W	deal with erroneous, noisy and missing data

4.2.2 Alternation with Motion Constraints

Focusing on the SFM problem, an *Alternation with motion constraints* is proposed in this Chapter. It is a variant of the Alternation similar to the one mentioned above for the special case of SFM [54]. In that approach, they impose that the last row of S should be equal to $\mathbf{1}$. In addition to that, the fact that M contains the relative camera-object motion at each frame is used. Therefore, given M_{k-1}, and before computing S_k, the orthonormality of the camera axes at each frame (namely the rows of M, taking only the first three columns) is imposed. The obtained algorithm is summarized below:

Alternation with motion constraints algorithm: The algorithm starts with an initial random $4 \times p$ matrix S_0, setting its last row to $\mathbf{1}$. The next steps are repeated until the product $M_k S_k$ converges to $W_{2f \times p}$:

1. Compute the matrix M_k:

$$M_k = W S_{k-1}^t (S_{k-1} S_{k-1}^t)^{-1} \tag{4.2}$$

   Define $M = [\ R \quad \mathbf{t}\ ]$, where R is a $2f \times 3$ matrix that contains the relative camera-object orientation at each frame, whereas $\mathbf{t}$ is a $2f \times 1$ vector that contains the relative camera-object position at each frame; which is the same, R and $\mathbf{t}$ are the rotation and translation components of the camera, respectively.

2. Impose orthonormality of camera axes, as in the *normalization* step in Section 2.1.1 (see equation (2.19)):

$$\begin{aligned} \mathbf{i}_i^t A A^{-1} \mathbf{i}_i &= 1 \\ \mathbf{j}_i^t A A^{-1} \mathbf{j}_i &= 1 \\ \mathbf{i}_i^t A A^{-1} \mathbf{j}_i &= 0 \end{aligned} \tag{4.3}$$

   where $\mathbf{i}_i$ and $\mathbf{j}_i$ are the x and y camera axes at frame i (see equation 2.7), and $i = 1, \cdots, f$, being f the number of frames.

   Actualize M_k:

$$\widetilde{M_k} = M_k A \tag{4.4}$$

3. Subtract to each column j of W (denoted as $\mathbf{w}^j$) the translation component of $\widetilde{M}_k$:

$$\widetilde{\mathbf{w}}^j = \mathbf{w}^j - \widetilde{\mathbf{t}}_k, \;\; j = 1, \ldots, p \tag{4.5}$$

   Compute the matrix $S_k = [\;(S_R)_k \quad \mathbf{1}\;]$:

$$(S_R)_k = ((R)_k^t (R)_k)^{-1}((R)_k^t \widetilde{W}) \tag{4.6}$$

4. Stop if the product $M_k S_k$ converges to W.

   Otherwise, set $k = k + 1$ and go to step 1.

Solution: The product $M_k S_k$ is the closest rank-r matrix to W, in the least-squares sense.

Due to the motion constraints added at each iteration, this Alternation variant provides an Euclidean 3D reconstruction of the object, instead of an affine one.

Adding prior knowledge

Furthermore, the addition of priors to the formulation by using known information is proposed. Other approaches remark the importance of adding priors in the SFM problem formulation. For instance, Gruber and Weiss [45] use the fact that the camera location at time $t + 1$ is probably similar to its location at time t. They show how the performance of their method is improved by adding this *temporal coherence* prior. Buchanan and Fitzgibbon [23] also add priors on the formulation. Concretely, they penalize the non-orthonormality of the camera in the optimization function. They show improvements in the results when this information is added.

In the *Alternation with motion constraints* presented above, the othonormality of the camera is imposed at the normalization step. However, camera motion is not considered. In sequences where the camera pose does not vary so much, it can be assumed, as in [45], that the camera has a *smooth* motion. To do that, a constraint that fix the rotation of the camera between consecutive frames, can be added in the normalization step. Since the camera axes are unitary vectors, the scalar product between consecutive camera axes give the *cosine* of the angle that they form. Setting a value for that *cosine*, the following equations are obtained:

$$\begin{aligned} \mathbf{i}_i^t A A^{-1} \mathbf{i}_{i+1} &= cos(\alpha) \\ \mathbf{j}_i^t A A^{-1} \mathbf{j}_{i+1} &= cos(\alpha) \end{aligned} \tag{4.7}$$

where $i = 1, \cdots, f-1$, f is the number of frames and α is the estimated angle between consecutive camera axes. These $2f-2$ equations should be added to the linear system generated by equations (4.3) to compute the matrix A.

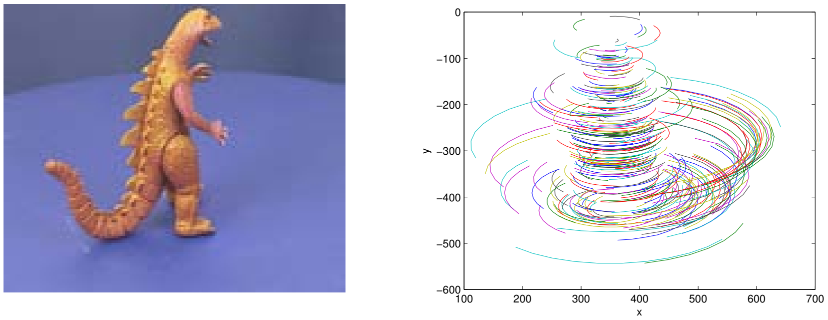


Figure 4.1: (left) first frame of the sequence; (right) initially known trajectories plotted in the image plane.

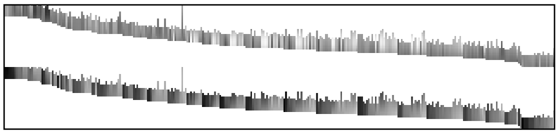


Figure 4.2: matrix of trajectories, only the 23.08% of data is known.

Unfortunately, this constraint can only be used when it is known that the camera has a smooth motion and, even in those cases, it is not easy to set α properly.

Example

The aim at introducing this example is to show that in particular sequences, to impose the smooth camera motion constraints can improve the obtained results.

A well-known real sequence, previously considered in Chapter 1, is used in this example. It is publicly available at: http://www.robots.ox.ac.uk/∼vgg/data/data-mview.html. The first frame of the sequence is shown in Fig. 4.1 (left). The matrix of trajectories W contains 319 feature point trajectories along 36 frames. That is, it has a dimension of 72×319 and the percentage of known data is only 23.08%. Fig. 4.1 (right) shows the trajectories plotted in the image plane. The matrix of trajectories is shown in Fig. 4.2. White entries correspond to missing data.

Fig. 4.3 shows an example of the recovered factors obtained with the Alternation technique. It should be remarked that the reported result has been selected from 1000 attempts: this is the best obtained result among 1000 attempts. With such amount

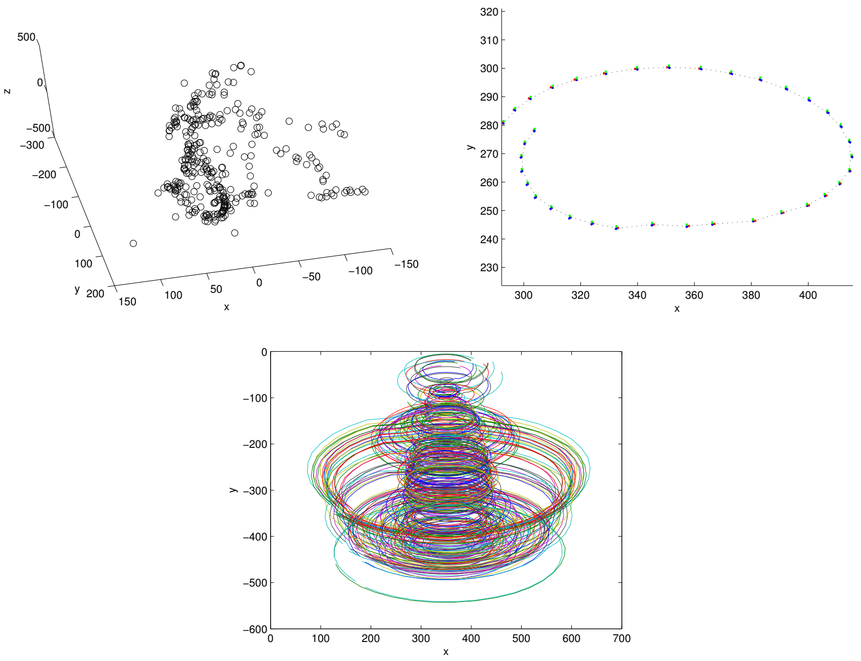


Figure 4.3: (top) recovered factors: 3D shape (left) and motion (right); (bottom) filled trajectories obtained by multiplying the recovered factors M and S, plotted in the image plane.

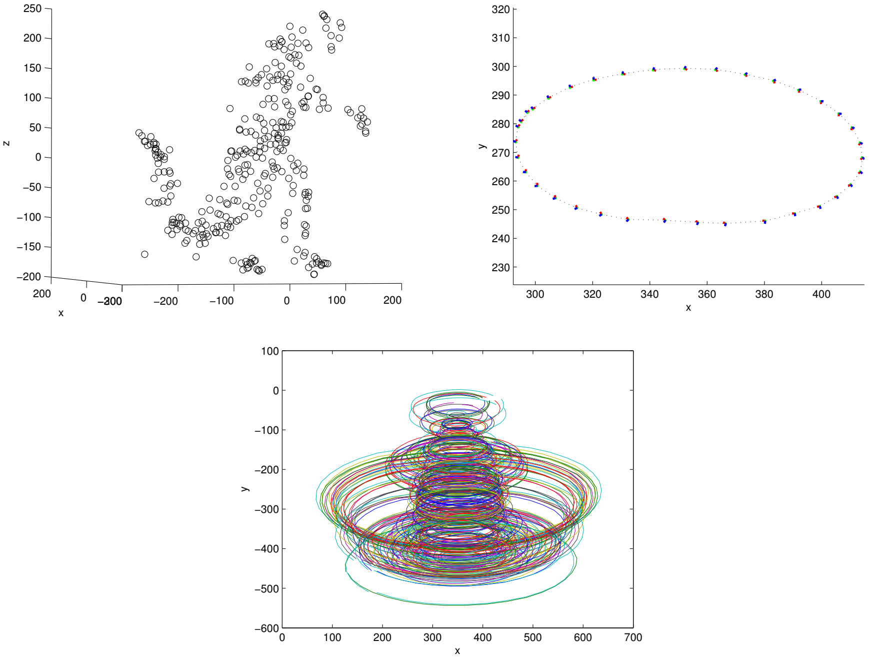


Figure 4.4: adding smooth camera motion constraints; (top) recovered factors: 3D shape (left) and motion (right); (bottom) filled trajectories obtained by multiplying the recovered factors M and S, plotted in the image plane.

of missing data, it is not easy to obtain a proper result with the Alternation. The goodness of the result is measured by using the *root mean square error* (*rms*), defined in equation (3.21), in Chapter 3. In this particular attempt, $rms = 1.3905$. It can be seen that some recovered trajectories are not closed. In fact, the recovered camera motion does not presents a closed trajectory.

Better results are obtained if smooth motion is imposed to the camera, as it can be seen in Fig. 4.4. This result corresponds to the selected attempt in the previous case. In this case, the *rms* value is 1.3732. The assumed *cosine* value in equations (4.7) is 0.98, which means an angle of about 10 degrees between consecutive frames. It can be appreciated that, in this case, the recovered 3D is more correct and the camera motion and the trajectories are better recovered.

4.3 Iterative Multiresolution Scheme

This Section presents the iterative multiresolution scheme proposed to fill in missing data. Essentially, the basic idea of the proposed approach is to work with submatrices that have a reduced percentage of missing data. Then, a factorization technique is used to obtain the 3D shape S and motion M of each of these submatrices and the missing data are filled in with the resulting product MS. In particular, the *Alternation with motion constraints* introduced in the previous Section is used. The proposed approach consists of two stages, which are described below.

4.3.1 Observation Matrix Splitting

Let $W_{2f \times p}$ be the matrix of trajectories (also referred to through this Chapter as original input matrix) of p feature points tracked over f frames containing missing entries; it will be denoted as W. Let k be the index indicating the current iteration number.

The aim at this first stage is to split the matrix of trajectories W in order to obtain submatrices with a reduced percentage of missing data. This splitting process consists of the following two steps:

- **Splitting:** in the first step, the original input matrix W is split into a set of $k \times k$ non-overlapped submatrices, defined as w_k^i, with a size of $\lfloor \frac{2f}{k} \rfloor \times \lfloor \frac{p}{k} \rfloor$ and where $i = 1, ..., k^2$. For the sake of presentation simplicity, hereinafter the input matrix at the current iteration level k will be referred to as W_k.
- **Multiresolution approach:** although the idea is to focus the process in a small area (submatrix w_k^i), which is supposed to have a reduced percentage of missing data, recovering information from a small patch can be affected from noisy data. Hence, in order to improve the confidence of recovered data, in this second step, and only when $k > 2$, a multiresolution approach is followed. It consists in computing four overlapped submatrices $\mathbf{w}_k^{i_n}$, $n = 1, ..., 4$ (see Fig. 4.5) with twice the size of w_k^i, and for every submatrix w_k^i.

  The idea of this enlargement process is to study the recovered entries in w_k^i when both different size regions are considered together with their overlapped results. The latter avoids obtaining results from small single regions. Other strategies were tested in order to compute in a fast and robust way submatrices with a lower percentage of missing entries (e.g., quadtrees, ternary graph structures), but they do not give the desired and necessary properties of overlapping.

  Since generating four $\mathbf{w}_k^{i_n}$, for every w_k^i, is a computationally expensive task, a simple and more direct approach is followed. It consists in splitting the matrix W_k in four different ways, by shifting $\mathbf{w}_k^{i_n}$ half of its size (i.e., w_k^i) through rows, columns or both at the same time. Fig. 4.6 illustrates the five partitions of matrix W, at the sixth iteration—i.e., the one generated by all the w_k^i and the other four ones, obtained with all the $\mathbf{w}_k^{i_n}$ submatrices, generated at this sixth iteration. When all these matrices are considered together, the overlap between the different areas is obtained, see textured cell in Fig. 4.5 and Fig. 4.6.

  Missing data at corners cells are only considered to be filled twice (w_k^i and one $\mathbf{w}_k^{i_n}$), while border cells three times (w_k^i and two $\mathbf{w}_k^{i_n}$). Other missing data in other cells are considered five times (w_k^i and its four $\mathbf{w}_k^{i_n}$).

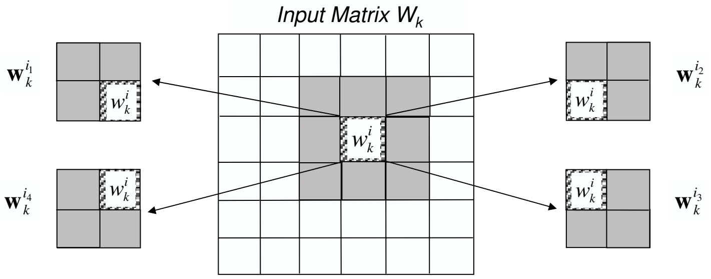


Figure 4.5: w_k^i and its four corresponding $\mathbf{w}_k^{i_n}$ matrices, computed during the first stage, at iteration $k = 6$.

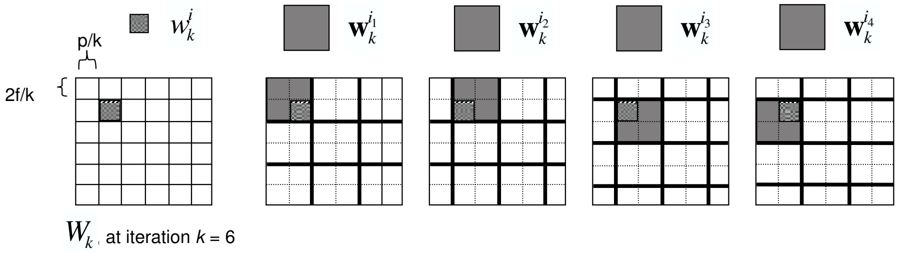


Figure 4.6: five partitions of matrix W_k. Note the overlap between a w_k^i submatrix with its corresponding four $\mathbf{w}_k^{i_n}$ submatrices, computed during the first stage.

4.3.2 Submatrices Processing

The objective at this stage is to recover missing data by applying an imputation technique at every single submatrix. At the same time, initially known values could also be modified. One important point that must be highlighted is that submatrices with a high percentage of missing data are discarded (as mentioned above, in the current implementation only submatrices with a percentage of missing data below 50% are considered).

As mentioned above, the *Alternation with motion constraints*, introduced in Section 4.2, is used to factorize the submatrices. Other methods to decompose the matrix of trajectories could be used. Actually, in [69], we compare the use of the *Alternation with the Damped Newton* technique [23] in the proposed multiresolution scheme and show that the first is more appropriated, both from the results and from the computational cost.

Independently of their size hereinafter submatrices will be referred to as W_s. Therefore, given a submatrix W_s with a percentage of known data of at least 50%, its corresponding M_s and S_s matrices are obtained by using the Alternation technique. Then, the product M_sS_s is used to fill in the matrix W_s. Finally, the root mean squared (rms) is computed as follows:

$$rms_s = \frac{\|W_s - M_sS_s\|_F}{\sqrt{n}} = \sqrt{\frac{\sum_{i,j} |(W_s)_{ij} - (M_sS_s)_{ij}|^2}{n}} \tag{4.8}$$

where i and j correspond to the index pairs where $(W_s)_{ij}$ is defined and n is the number of those pairs in W_s.

Since the rms_s is generally adopted as a measure of the goodness of the recovered data, it will be used later on as a weighting factor for merging data on overlapped areas after finishing the current iteration. Concretely, every point of the filled W_s is associated with a weight, defined as $\frac{1}{rms_s}$.

In the original approach [68] and [69], a threshold ς was defined and used to discard the recovered entries in W_s when its corresponding rms_s was higher than ς. The problem was how to tune that threshold. One of the main improvements of the new formulation proposed in this Chapter is that, since only matrices with a reduced percentage of missing data are used, no ς–threshold is needed to be defined.

Finally, when every submatrix W_s has been processed, recovered missing data are used for filling in the original input matrix W. In case a missing entry has been recovered from more than one submatrix (overlapped regions) those recovered data are merged by using their corresponding normalized weighting factors. The average of the initial value and the new one, obtained after the merging process, is assigned.

Once recovered missing data have been used for filling in the input matrix W, the iterative process starts again (Section 4.3.1), splitting the new matrix W either by increasing k by one or, in case the size of submatrices w_k^i at the new iteration stage is too small, by setting $k = 2$. This iterative process is applied until one of the following conditions is true: a) a maximum number of iterations is reached; b) at the current iteration no missing entries were recovered; c) the matrix of trajectories is totally filled. An outline of the algorithm can be found below; Fig. 4.7 presents an overview of the algorithm.

Iterative multiresolution Scheme's Algorithm

Inputs: W original input matrix of trajectories; *data*: percentage of known data in W; *itmax*: maximum number of iterations; *minsize*: submatrix minimum size.

Set $k = 2$, $it = 1$, $W_0 = W$ and repeat the following steps while: $(it < itmax)$ and $(\ data_k > data_{k-1})$ and $(\ data_k < 100\%)$

1. Split the matrix W_0 into $k \times k$ submatrices w_k^i, obtaining W_k.

   If $size(w_k^i) < minsize$, set $k = 2$, $it = it + 1$ and repeat step 1.

2. Multiresolution approach: compute the four partitions of matrix W_k (generated by $\mathbf{w}_k^{i_n}$, $n = 1, ..., 4$).

3. Submatrices processing: apply *Alternation with motion constraints* to all the submatrices.

4. Merge the data by using the weights and update W_k. Set $W_0 = W_k$, $k = k+1$, $it = it + 1$. Go to step 1.

Solution: $W_{filled} = W_k$, $data_k > data_0$

4.4 Evaluation Study

The aim at this stage is to study the robustness to missing and noisy data of a factorization technique applied to the partially or totally filled matrix obtained with the proposed iterative scheme. This study is performed by comparing the result when the same factorization technique is applied directly to the original input matrix. In summary, the methodology proposed to evaluate the obtained results, which is shown in Fig. 4.8, consists in applying:

- a factorization technique over the original input matrix W;
- a factorization technique over the matrix filled in with the iterative multiresolution scheme W_{filled}.

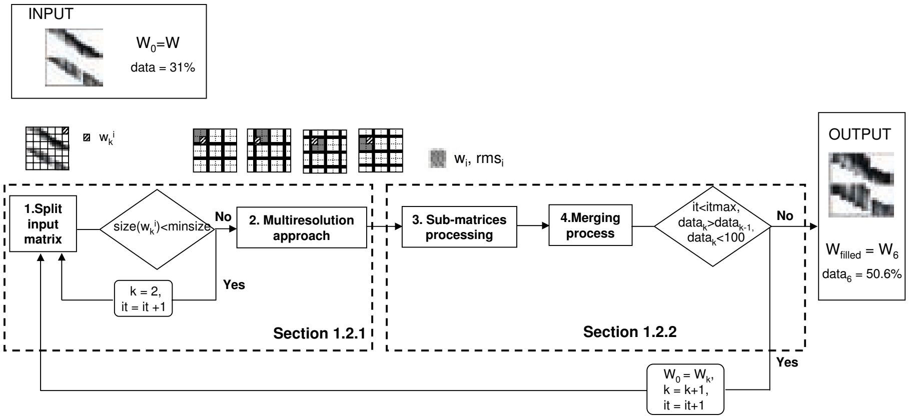


Figure 4.7: scheme of the algorithm; example of an input matrix W with only 31% of known data. The obtained filled matrix W_{filled} after 6 iterations contains 50.6% of known data.

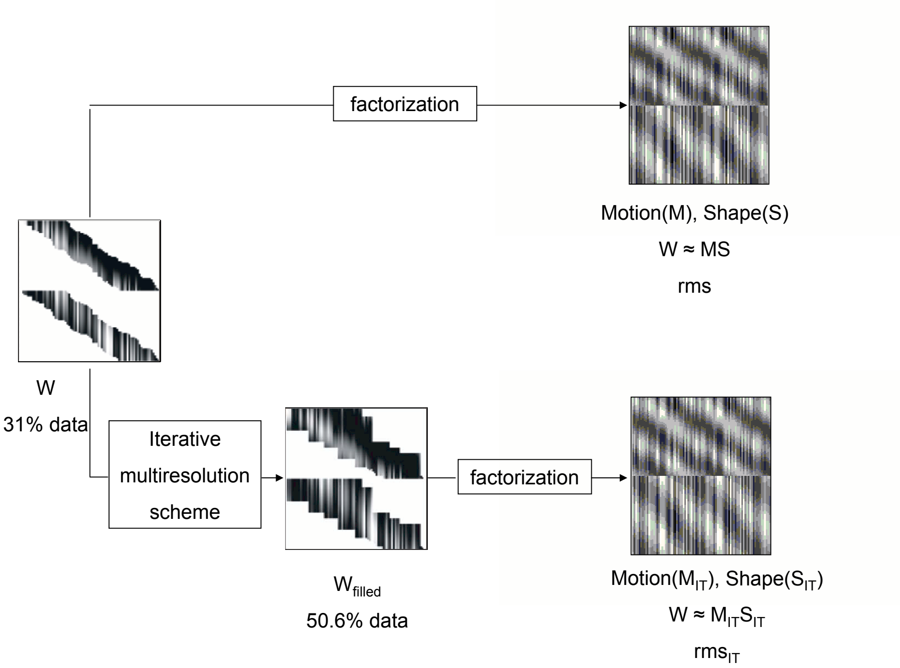


Figure 4.8: evaluation study; the recovered factors and the obtained *rms* in each case are compared. In our experiments, the *Alternation with motion constraints* is used as factorization technique. IT means iterative scheme. In this example, the input matrix has only 31% of known data.

Although any factorization technique could be tested, the *Alternation with motion constraints* technique is used in the evaluation study. Actually, this technique is also used inside the proposed scheme to fill in missing entries of submatrices. Since the objective is to study the performance of the proposed iterative multiresolution scheme, the *smooth camera motion* constraints are not imposed in the experimental results. For the shake of simplicity, and only in the current Chapter, *Alternation with motion constraints* is referred to as Alternation, hereinafter.

Experiments using both synthetic and real data are presented below. Different amounts of missing data are considered—from 10% up to 70%. Furthermore, different levels of Gaussian noise—standard deviation σ with a value from $\frac{1}{3}$ to 1 and zero mean—are added to the 2D feature point trajectories, for the synthetic case. The obtained matrices are denoted as $\widehat{W}$. Notice that in the case of real data the original input matrix W already contains noisy values. For each setting (amount of missing data and level of noise) 100 attempts are repeated and the root mean square error (rms) is computed:

$$rms = \frac{\|\widehat{W} - MS\|_F}{\sqrt{n}} = \sqrt{\frac{\sum_{i,j} |(\widehat{W})_{ij} - (MS)_{ij}|^2}{n}} \tag{4.9}$$

where i and j correspond to the index pairs where $(\widehat{W})_{ij}$ is defined and n is the number of those pairs in $\widehat{W}$.

Furthermore, and also in the synthetic case, the recovered M and S are compared to the ground truth matrices (M_G and S_G) with the strategy explained below (see 4.4.1).

Given a matrix where all values are known W_{all}, different percentages of missing data are generated by automatically removing parts of random columns in order to simulate the behaviour of tracked features. The removing process selects randomly a cell in the given column, dividing it into two parts. One of these parts is randomly removed, simulating features missed by the tracker or new features detected after the first frame, respectively. Different numbers of frames could be used to achieve the percentages of missing data, but the idea is to work with matrices of the same size, since the performance of factorization techniques is not independent of the size of the matrix. Note that missing data could simply be obtained by randomly removing entries in W_{all}, but it would not simulate a realistic situation. Furthermore, Alternation technique performs better when a random missing data distribution is considered (see Section 3.3) and it may not be appropriated for an evaluation study.

As pointed out in [24], the rms defined by the expression (4.9) could be ambiguous and in some cases contradictory. That is because it only takes into account the recovered values corresponding to initially known entries in the original input W, but it ignores how the rest of entries are filled in. Since in our experiments all the entries are initially known in W_{all}, as mentioned above, the root mean square error considering all the entries in W_{all} can be computed. Hereinafter, this measure will be referred to as rms_{all} and it is defined as follows:

$$rms_{all} = \frac{\|W_{all} - MS\|_F}{\sqrt{2fp}} \tag{4.10}$$

where $2fp$ is the size of the matrix W_{all}.

4.4.1 Motion and Shape Error Computation

One possibility to study the goodness of the recovered shape S and motion M matrices consists in expressing the rotations relative to that of the first frame, both the ground truth rotations and the recovered ones, as suggested in [110]. However, first of all, the coordinate system must be at the same position. Therefore, the computed translation vector $\mathbf{t}$, is transformed so that coincides with the real one $\mathbf{t}_G$:

$$\widetilde{\mathbf{t}} = \mathbf{t}_G - \mathbf{t}; \tag{4.11}$$

Then, the camera pose at each frame i is expressed relative to that of the first frame:

$$\widetilde{\mathtt{R}}_i = \mathtt{R}_i \mathtt{R}_1^{-1} \tag{4.12}$$

$$\widetilde{\mathtt{R}}_{G_i} = \mathtt{R}_{G_i} \mathtt{R}_{G_1}^{-1} \tag{4.13}$$

where $i = 1, \cdots, f$, being f the number of frames, and:

$$\mathtt{R}_i = \begin{bmatrix} \mathbf{i}_i^t \\ \mathbf{j}_i^t \\ \mathbf{i}_i^t \times \mathbf{j}_i^t \\ \end{bmatrix} \tag{4.14}$$

is the computed camera pose at frame i. Analogously for the ground truth camera pose $\mathtt{R}_{Gi}$. These matrices and the translation components $\widetilde{\mathbf{t}}$ and $\mathbf{t}_G$ are used to construct the aligned motion matrices $\widetilde{M}$ and $\widetilde{M}_G$, respectively.

Finally, the first three rows (the last is equal to $\mathbf{1}$) of the recovered S and of the ground truth S_G should be also aligned:

$$\widetilde{S} = \mathtt{R}_1 S \tag{4.15}$$

$$\widetilde{S}_G = \mathtt{R}_{G_1} S_G \tag{4.16}$$

The following expressions are used to measure the goodness of the recovered factors:

$$rms_S = \frac{\|\widetilde{S}_G - \widetilde{S}\|_F}{\sqrt{3p}} \tag{4.17}$$

$$rms_M = \frac{\|\widetilde{M}_G - \widetilde{M}\|_F}{\sqrt{3(2f)}} \tag{4.18}$$

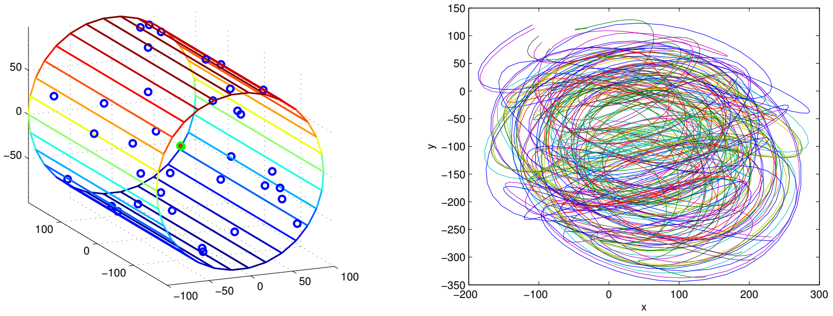


Figure 4.9: synthetic object; (left) cylinder; (right) feature point trajectories represented in the image plane.

being S and M of sizes $4 \times p$ and $2f \times 4$, respectively. However, since the last row of $\widetilde{S}_G$ and $\widetilde{S}$ are equal (both contain a vector of ones) and the fourth column of $\widetilde{M}_G$ and $\widetilde{M}$ are also identical (both translation vectors are equal), the corresponding entries are not taken into account when computing the number of elements (see the denominator in each rms expression).

4.4.2 Synthetic Data

This Section provides results obtained with two data sets from different objects. The first data set is generated by randomly distributing 3D feature points over the surface of a cylinder, see Fig. 4.9 (left). The second data set is generated from 3D points of a triangular mesh (nodes), representing a Beethoven sculptured surface, see Fig. 4.17 (left). In this second object the points are not as uniformly distributed as in the previous object. Two different sequences are obtained with these objects by performing a rotation and a translation over each of them. At the same time, the camera also rotates and translates. Although missing data can be obtained due to self occlusions of the objects, all the points are considered, as mentioned above.

The first sequence is defined by 200 frames containing 300 features. The trajectories are plotted in Fig. 4.9 (right). Fig. 4.10 shows an example of recovered shape (left) and motion ((middle) and (right)) obtained by applying the Alternation technique to the matrix filled with the proposed iterative scheme. The original input matrix has about 20% of missing data.

The obtained error in each attempt, considering different percentages of missing data, is shown. The reported experiments correspond to the no noise case and to cases with added Gaussian noise of a *standard deviation* (σ) of 1/3 and 1, respectively.

Fig. 4.11 (left) shows the obtained rms (in logarithmic scale) for the case of no noise. Recall that polygonal figures enclose data in between lower and upper quartiles (medians are represented by horizontal lines in thinner regions). Vertical lines, outside these polygonal figures, correspond to the rest of data. It can be seen that, when the

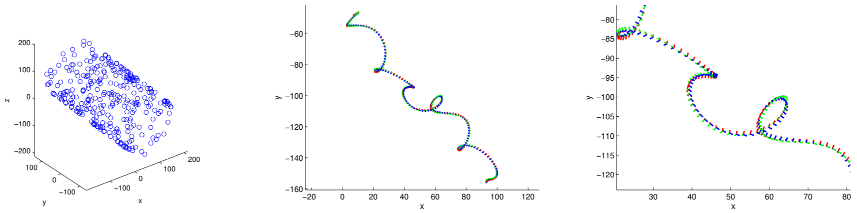


Figure 4.10: cylinder scene; (left) 3D reconstruction: notice that there are thicker points which correspond to the reappearing features; (middle) recovered camera motion; (right) enlargement of the recovered camera motion.

percentage of missing data is below 50%, no improvements are obtained by using the iterative scheme, since the Alternation performs quite well with such amount of missing data. With more missing data, the Alternation applied to the matrix filled in with the iterative scheme performs better than applied directly to the original input matrix W.

As mentioned above, the rms_{all}, which considers all the entries in the original input matrix instead of only the initially known ones, is also studied. Fig. 4.11 (right) shows the obtained values in the case of no noise in the data. Notice that when the Alternation is applied directly to the input matrix, the rms_{all} takes higher values than the rms when the percentage of missing data is equal or higher than 50%. This is due to the fact that with such amount of missing data, a global solution is not always found and the error corresponding to initially unknown entries is higher than the error corresponding to initially known ones.

The obtained results when a Gaussian noise with $\sigma = 1/3$ and $\sigma = 1$ is added to the data are plotted in Fig. 4.12 and Fig. 4.13, respectively. As in the case of no noisy data, the Alternation applied to the matrix filled in with the iterative scheme performs better than applied directly to the original input W, while the percentage of missing data is equal or higher than 50%. It can be seen that the difference between the error obtained by applying Alternation to the filled in matrix with the iterative scheme and to the original one is not as significant as in the free noise case.

The recovered factors S and M are also studied. The error is measured by computing the rms_S (equation (4.17)) and rms_M (equation (4.18)), which are plotted in Fig. 4.14 (no noise case), Fig. 4.15 ($\sigma = 1/3$ case) and Fig. 4.16 ($\sigma = 1$ case). It can be seen that results are similar to the ones obtained with the rms: better results are obtained when Alternation is applied to the matrix filled in with the iterative scheme, while the percentage of missing data is equal or higher than 50%.

In the second sequence, the numbers of frames and trajectories considered are 200 and 266, respectively. Feature point full trajectories are plotted in Fig. 4.17 (right). Fig. 4.17 (left) contains a large amount of 3D points (about 2655) in order to visualize better the object. Fig. 4.18 shows an example of the recovered shape (left) and motion ((middle) and (right)) obtained by applying Alternation to the matrix filled in with the proposed iterative scheme (input matrix with 20% of missing data).

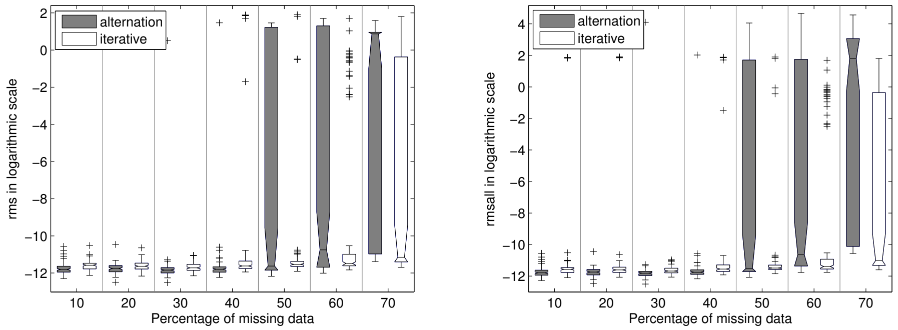


Figure 4.11: cylinder scene, no noise case; (left) *rms* in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

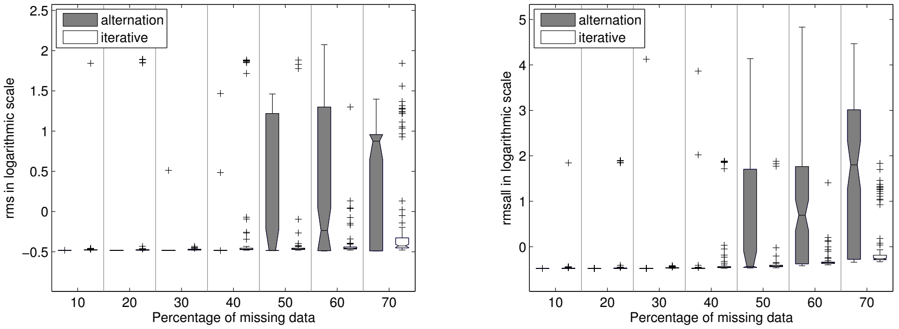


Figure 4.12: cylinder scene, $\sigma = 1/3$ case; (left) *rms* in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

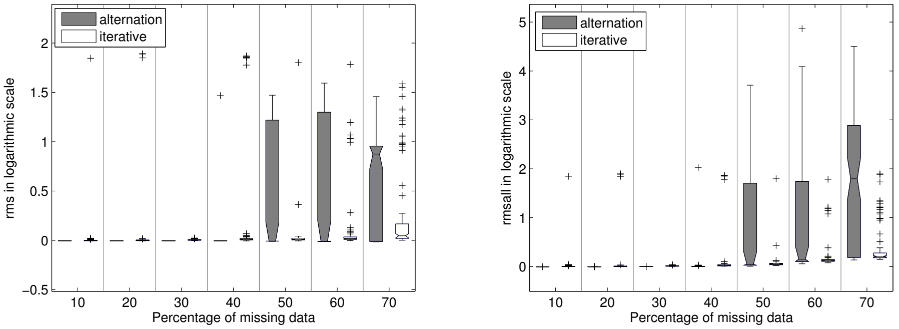


Figure 4.13: cylinder scene, $\sigma = 1$ case; (left) *rms* in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

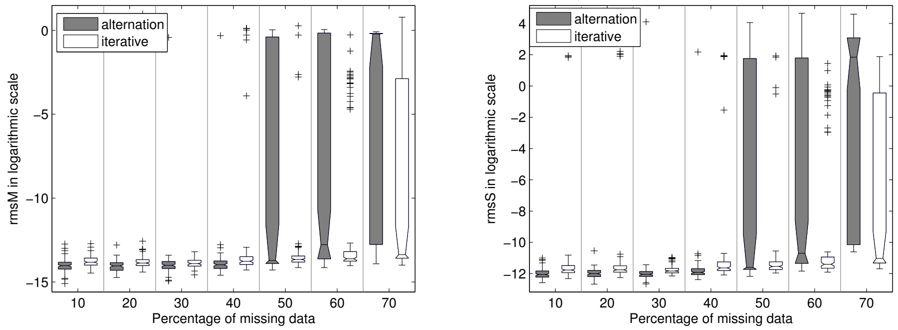


Figure 4.14: cylinder scene, no noise case; (left) rms_M in logarithmic scale, for different percentages of missing data; (right) rms_S in logarithmic scale, for different percentages of missing data.

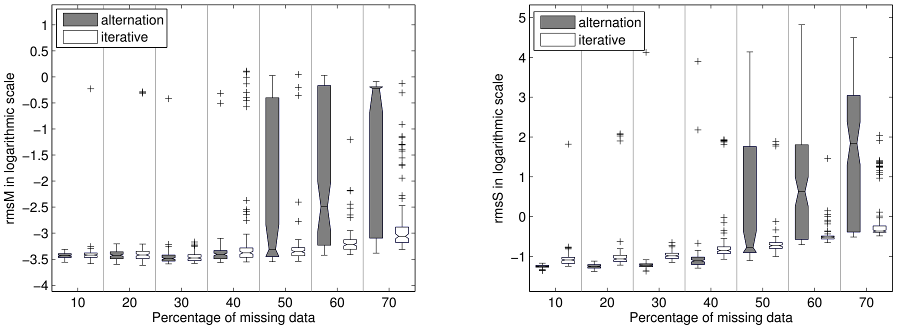


Figure 4.15: cylinder scene, $\sigma = 1/3$ case; (left) rms_M in logarithmic scale, for different percentages of missing data; (right) rms_S in logarithmic scale, for different percentages of missing data.

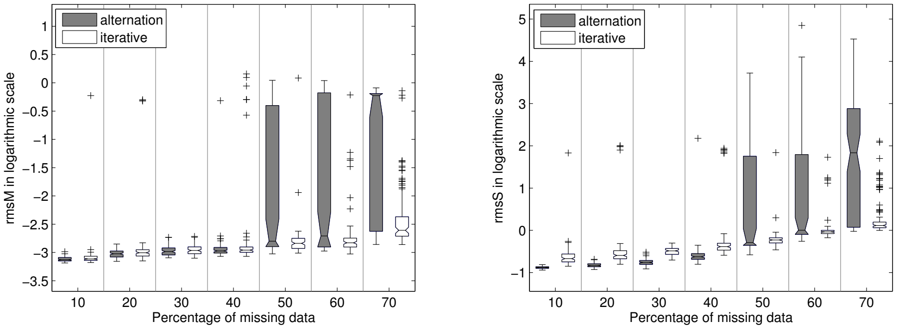


Figure 4.16: cylinder scene, $\sigma = 1$ case; (left) rms_M in logarithmic scale, for different percentages of missing data; (right) rms_S in logarithmic scale, for different percentages of missing data.

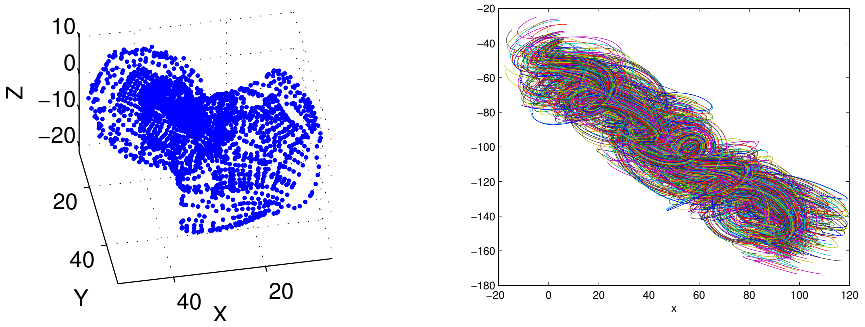


Figure 4.17: synthetic object; (left) Beethoven's sculpture; (right) feature point trajectories represented in the image plane.

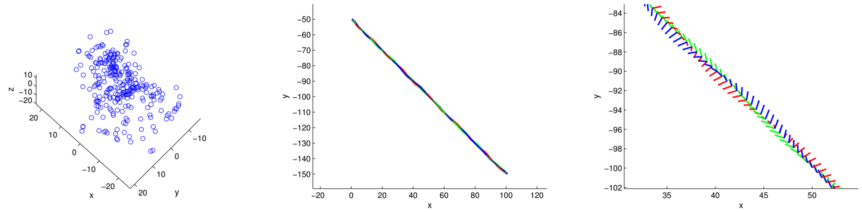


Figure 4.18: Beethoven's sculpture scene; (left) 3D reconstruction; (middle) recovered camera motion; (right) an enlargement of the recovered camera motion.

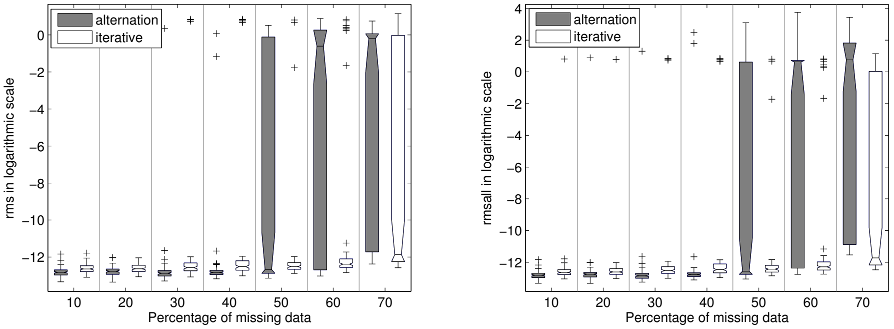


Figure 4.19: Beethoven's sculpture scene, no noise case; (left) rms in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

The obtained rms and rms_{all}, in this second sequence are plotted in Fig. 4.19 (no noise case), Fig. 4.20 ($\sigma = 1/3$) and Fig. 4.21 ($\sigma = 1$). In the case of no noisy data (Fig. 4.19), results are similar to the previous sequence: the Alternation gives smaller rms applied to the matrix filled in with the proposed iterative scheme than to the original input one, while the percentage of missing data is equal or higher than 50%. It can be seen that, for percentages of missing data below 50%, results are similar in both cases.

When working with noisy data, for the particular case of $\sigma = 1/3$, the rms is smaller when the Alternation is applied to the matrix filled in with the iterative scheme, while the amount of missing data is equal or higher than 50%. However, the difference is not as significant as in the no noise case. For the case of $\sigma = 1$, the obtained rms applying the Alternation to the input matrix are very similar to the ones obtained applying the Alternation to the matrix filled in with the iterative scheme. Only in the case of 60% of missing data the rms is smaller when the Alternation is applied to the filled in matrix with the proposed iterative scheme. The rms_{all} is smaller when Alternation is applied to the filled in matrix, while the percentage of missing data is equal or higher than 60%.

The rms_M and rms_S are shown in Fig. 4.22 (no noise case), Fig. 4.23 ($\sigma = 1/3$) and Fig. 4.24 ($\sigma = 1$). In the three cases, both errors are smaller when the Alternation is applied to the matrix filled in with the iterative scheme, while the percentage of missing data is equal or higher than 50%. Therefore, although in the case of $\sigma = 1$ the obtained rms and rms_{all} were similar applying Alternation to the input matrix and to the filled in one with the iterative scheme (see Fig. 4.21), the rms_S and rms_M are smaller in the second case.

As a conclusion, it can be said that in general, the Alternation applied to the original input matrix performs quite well, while the percentage of missing data is small (below 50%). Therefore, it is not necessary to firstly apply the iterative multiresolution scheme in those cases. However, the results obtained with the Alternation get

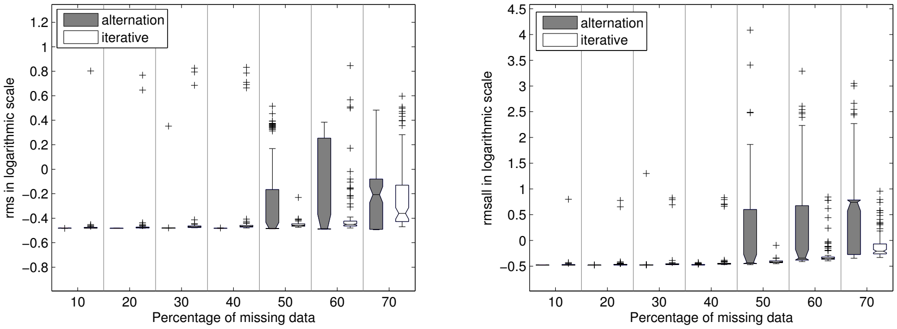


Figure 4.20: Beethoven's sculpture scene, $\sigma = 1/3$ case; (left) rms in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

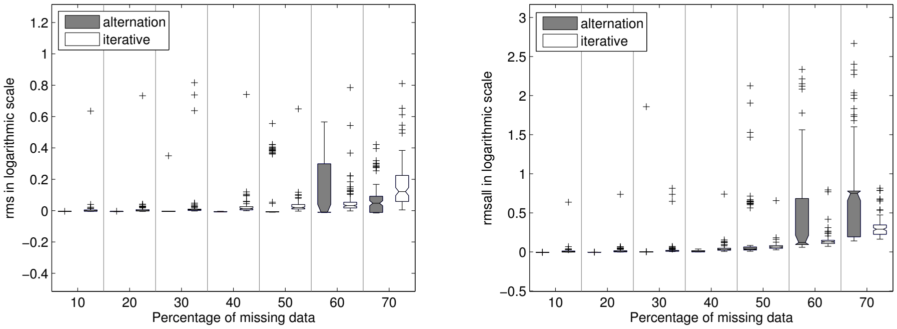


Figure 4.21: Beethoven's sculpture scene, $\sigma = 1$ case; (left) rms in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

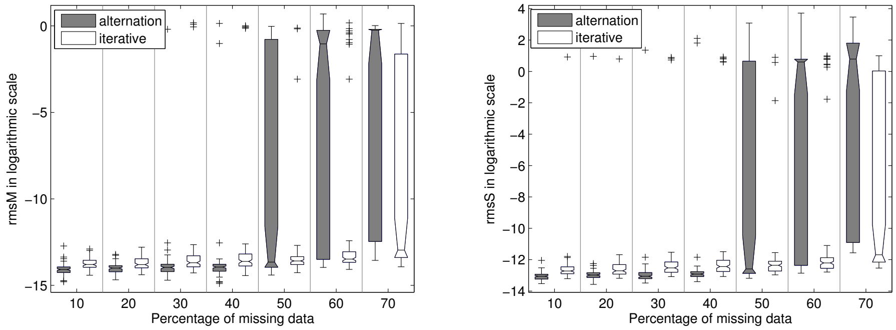


Figure 4.22: Beethoven's sculpture scene, no noise case; (left) rms_M in logarithmic scale, for different percentages of missing data; (right) rms_S in logarithmic scale, for different percentages of missing data.

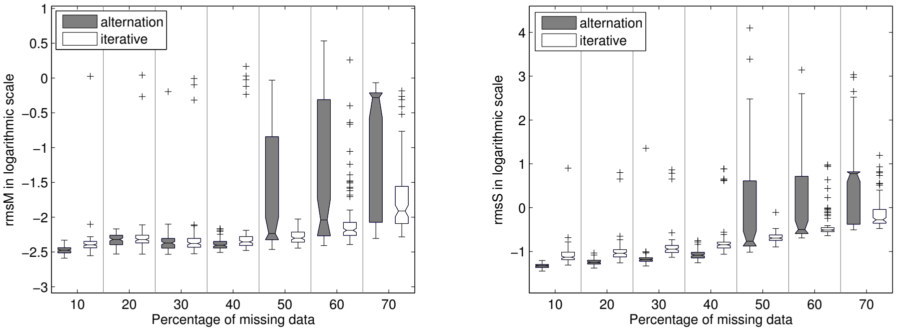


Figure 4.23: Beethoven's sculpture scene, $\sigma = 1/3$ case; (left) rms_M in logarithmic scale, for different percentages of missing data; (right) rms_S in logarithmic scale, for different percentages of missing data.

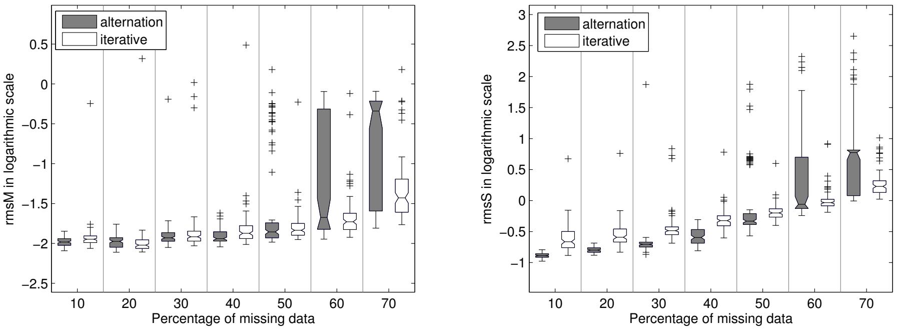


Figure 4.24: Beethoven's sculpture scene, $\sigma = 1$ case; (left) rms_M in logarithmic scale, for different percentages of missing data; (right) rms_S in logarithmic scale, for different percentages of missing data.

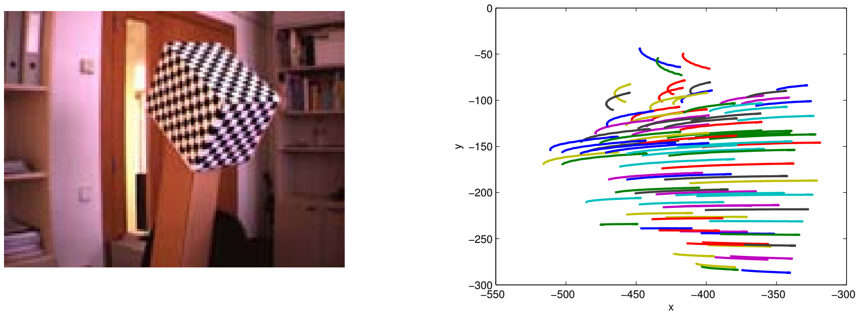


Figure 4.25: (left) first object used for the real scene; (right) feature point trajectories represented in the image plane.

worse as the percentage of missing data grows. Actually, the number of cases in which the Alternation applied to the original input matrix converges to a local minimum increases as the percentage of missing data grows. Hence, when the percentage of missing data is high, it is better to apply the proposed multiresolution scheme as a previous step, in order to reduce the percentage of missing data in W. The reported results show that the Alternation applied to this partially or totally filled in matrix gives better results than applied to the original input matrix.

4.4.3 Real Data

A procedure similar to the one applied to the synthetic data is now used with real data. The two objects studied in these real data experiments are shown in Fig. 4.25 (left) and Fig. 4.28 (left). For each object, a real video sequence with a resolution of 640×480 pixels is used and a single rotation around a vertical axis is performed. Feature points are selected by means of a corner detector algorithm. An iterative feature tracking algorithm has been used. More details about corner detection and tracking algorithm can be found in [82]. As in the previous case, all the points are initially known in W_{all}, because only full trajectories are considered. Missing data are generated automatically by removing parts of random columns, as in the synthetic data experiment. In most of the cases, the error values are larger than in the synthetic case. The problem is that both objects do not rotate so much, as it can be seen in the plot of the trajectories (Fig. 4.25 (right) and Fig. 4.28 (right)). Hence, the obtained matrices of trajectories are not of full rank (4) and we have to deal with a degenerate case.

In the first sequence, 87 points distributed over the squared-face-box are tracked along 101 frames. Feature point trajectories are plotted in Fig. 4.25 (right). Fig. 4.26 shows an example of the recovered shape (left) and motion ((middle) and (right)) obtained by applying Alternation to the matrix filled in with the proposed iterative scheme. In this example, the original input matrix has only about 10% of missing data.

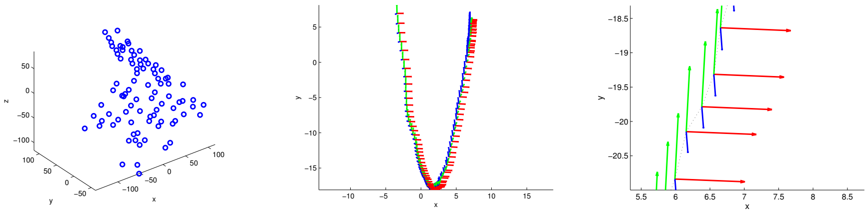


Figure 4.26: first object; (left) 3D reconstruction; (middle) recovered camera motion; (right) an enlargement of the recovered camera motion.

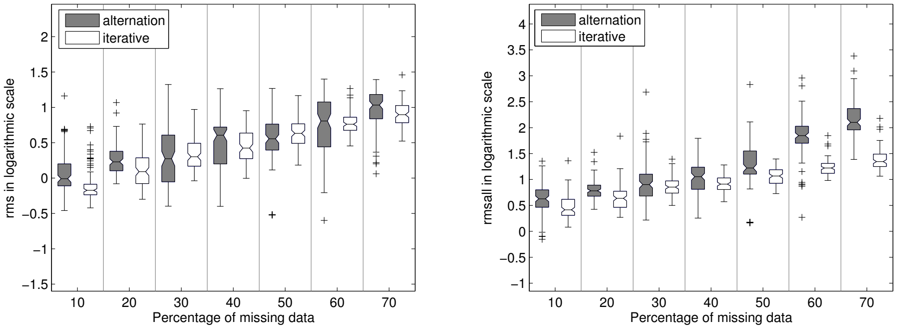


Figure 4.27: first object; (left) rms in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

The resulting rms obtained for different percentages of missing data are presented in Fig. 4.27 (left). It can be seen that the Alternation applied to the matrix filled in with the iterative scheme performs better than applied directly to the original input matrix, for any percentage of missing data. The rms_{all}, which takes into account all the entries in the matrix W, is plotted in Fig. 4.27 (right). As in the rms, the rms_{all} is smaller when the Alternation is applied to the matrix filled in with the iterative scheme. However, notice that the difference between applying the Alternation to the filled in matrix with the proposed scheme and to the input matrix is not as significant as in the synthetic case.

The second sequence consists of 61 frames and 188 feature points. Feature point trajectories are plotted in Fig. 4.28 (right). Fig. 4.29 shows an example of the recovered shape (left) and motion ((middle) and (right)) obtained by applying Alternation to the matrix filled in with the proposed iterative scheme. The original input matrix has only about 10% of missing data.

In this second object, the error values are higher than before, as it can be seen in Fig. 4.30. The rms (left) and the rms_{all} (right) are smaller when the Alternation is applied to the matrix filled in with the iterative scheme than when applied to the original input one, for any percentage of missing data.

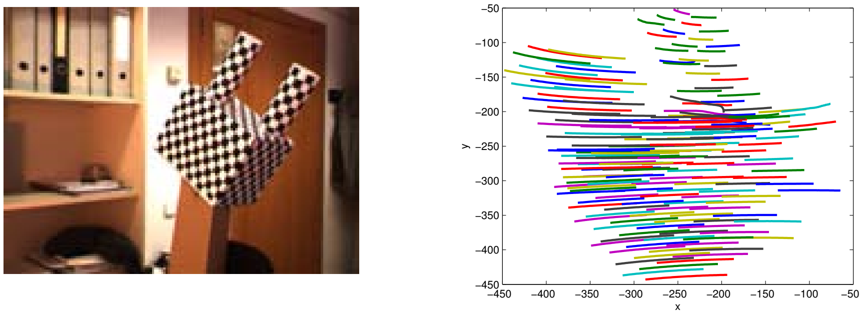


Figure 4.28: (left) second object used for the real scene; (right) feature point trajectories represented in the image plane.

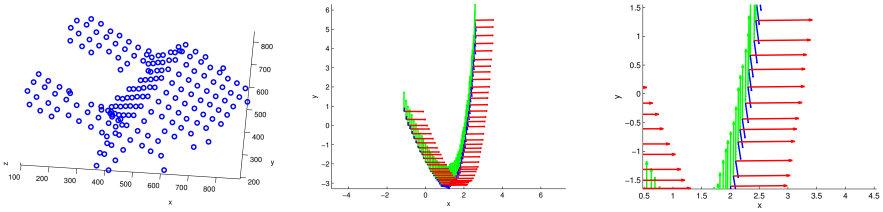
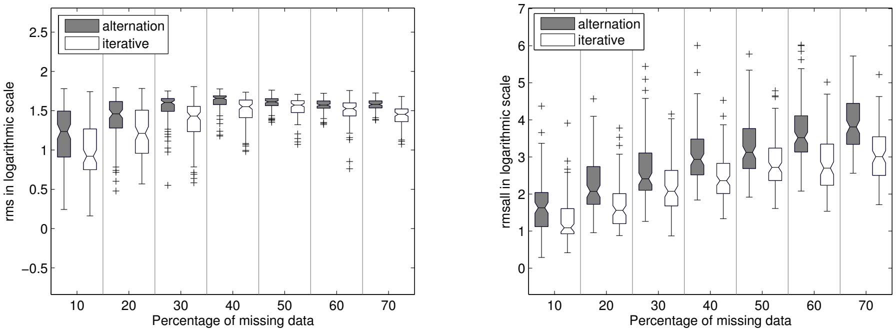


Figure 4.29: second object; (left) 3D reconstruction; (middle) recovered camera motion; (right) an enlargement of the recovered camera motion.



,

Figure 4.30: second object; (left) rms in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

As a conclusion from the real data experiments, it can be observed that the Alternation applied to the matrix filled in with the proposed iterative scheme gives better results than applied to the original input matrix, even when the percentage of missing data is low.

4.5 Summary

This Chapter presents an iterative multiresolution scheme for tackling the SFM problem when factorization techniques may fail due to a high ratio of missing data. The idea of the proposed iterative multiresolution scheme is to take submatrices of the input matrix with a low percentage of missing data, apply a factorization technique to obtain the shape S and motion M and recover the missing entries with the product MS. The *Alternation with motion constraints* technique is proposed and used to factorize these submatrices. The goal is to improve the results obtained when a factorization technique is applied to the matrix filled in with the iterative scheme instead of directly to the input one, which has a lower percentage of known data.

An evaluation study of the performance of the proposed scheme is given by using the *Alternation with motion constraints*. It has been shown that, when the percentage of missing data is high, the Alternation applied to the matrix filled in with the proposed iterative scheme gives better results than when applied directly to the original input matrix W. However, when the ratio of missing data is low, the Alternation performs quite well applied to the original input matrix directly and it is not necessary to use the iterative scheme. The goodness of the results is measured with the root mean square error (rms) and also with the rms_{all}, which takes into account all the entries in the initially full matrix W_{all}. In the synthetic case the recovered shape and motion matrices are also studied.

In summary, the proposed scheme can be used to obtain the shape and motion when the factorization techniques do not perform as well as expected due to the high percentage of missing data.

Chapter 5

SFM in the Multiple Objects Case with Missing Data

The Structure from Motion (SFM) problem with multiple objects is tackled in this Chapter. It is focused on the particular case of input matrix with missing data. The key point is firstly to segment the trajectories according to the motion of the objects (motion segmentation) and then, apply a single object SFM technique to the trajectories of each object.

An approach that consists of two stages is proposed to tackle the motion segmentation problem with missing data. In the first stage, the main objective is to fill in empty entries with an imputation method, without using or assuming a prior knowledge of the objects contained in the scene—number of objects and motion. Since the rank of the matrix of trajectories is needed to be known in order to apply the imputation method, a novel technique to estimate the rank of a missing data matrix is proposed. The intuition behind the proposed technique is that, since feature point trajectories belong to surfaces of rigid objects, the frequency spectra of the input matrix of trajectories should be preserved after recovering missing entries. In the second stage, motion segmentation is obtained by using a clustering approach based on the normalized cuts criterion.

Once the trajectories are segmented into different objects, any factorization technique for the single object case can be applied to obtain the Structure and Motion of each of the objects. Experiments with synthetic and real data are provided in order to empirically show the good performance of the proposed rank estimation technique and also the significance of a good rank estimation for correct motion segmentation results. Furthermore, the shape and motion of obtained clusters in the experiments are computed, both for correct cases and for cases where several features are wrongly segmented.

5.1 Introduction

In the single object case, factorization techniques are widely used to decompose the matrix of trajectories W into the shape and motion matrices. The decomposition is obtained by using the fact that the rank of the matrix of trajectories is at most 4, in the particular case of rigid objects and under affine camera model. Unfortunately, in the multiple objects case, factorization techniques can not be directly used to factorize a matrix of trajectories into the shape and motion matrices; on the one hand, each object has different motion and on the other hand, trajectories are not sorted into different objects in the matrix of trajectories. Working with an affine camera model, it can be used the fact that feature trajectories corresponding to the same object lie in the same linear subspace. Therefore, the aim is to find each of these linear subspaces in order to reduce the matrix of trajectories to a form that allows an easy identification of them. Hence, the *Structure from Motion* (SFM) problem in the multiple objects case is reduced to firstly segment the trajectories according to the motion of the objects (*motion segmentation*) and then, apply a single object SFM technique to the trajectories of each object. More details on the multiple objects formulation can be found in Section 2.1.2.

Another problem that should be faced out in the case of multiple objects is that the rank of the matrix W is unknown. Furthermore, its value is not bounded as in the single object case, since the number of objects and the kind of movement are not known. Many approaches proposed for the motion segmentation problem (e.g., [12, 26, 53, 72, 130, 126]) estimate the rank of the matrix of trajectories by studying the ratio between its singular values (obtained by applying the SVD, for instance). In [72], *model selection* techniques are proposed to estimate the rank of matrices with noisy data. The problem of model selection techniques is that there are parameters that depend on the amount of noise in the data, which are difficult to set properly without using information about the noise. Vidal and Hartley [117] propose to fix the rank of W to five and apply the SVD to project the point trajectories from $\mathbb{R}^{2f}$ to $\mathbb{R}^5$. Hence, it is not needed to estimate the rank of W. Then, the Generalized Principal Component Analysis (GPCA) [119], briefly presented in Section 2.2.4, is used to segment the trajectories. Tron and Vidal [115], presents a comparison of several motion segmentation algorithms. Two of the studied approaches are the ones presented in [117] and [126]. For the latter, the authors suggest to fix the rank of the matrix of trajectories instead of using model selection techniques. In particular, they compare the results obtained by fixing the rank to 5, which is the dimension used in [117], and to $4N$, where N is the number of motions in the scene.

In most of the aforementioned approaches it is assumed that tracked feature points are visible along the whole video sequence, giving rise to a full matrix of trajectories. However, trajectories are often incomplete or split up due to objects occlusions, missing on the tracking or simply because they exit the camera field of view. In this case, the rank of W can not be estimated as in the full data case, since the singular values of a missing data matrix can not be directly computed. The approach presented in [117] can deal with missing data. It proposes to fix the rank to 5 beforehand, for every matrix of trajectories and the PowerFactorization [54] is used to project the data into $\mathbb{R}^5$. Then, the GPCA is used for segmenting the trajectories. However, the performance of the GPCA tends to be worse as the number of motions in the scene increases. In addition, it is sensitive to outliers, as pointed out in [115].

This Chapter proposes an approach to tackle the motion segmentation problem with missing data in W. It consists of two stages. In the first stage, a novel technique to estimate the rank of a missing data matrix is presented. The proposed technique estimates the rank without using any prior knowledge of the scene (i.e., number of rigid objects, kind of motion). The intuition behind the proposed approach is that, since feature point trajectories belong to surfaces of rigid objects, the *frequency spectra* of the input matrix W should be preserved after recovering missing entries. Behind this idea there is the assumption that the behaviour of missing data and known data, studied as trajectories (columns of W), is the same; therefore, both generate the same spectral representation. The estimated rank is used for obtaining a full matrix by applying an imputation technique. In the second stage, an approach similar to the one presented in [126] is used to obtain the motion segmentation of the feature trajectories. Finally, the shape and motion of each of the objects can be recovered by using any SFM technique (e.g., [110]).

In order to show the significance of a good estimation of the rank for correct motion segmentation results, an approach similar to the one introduced in [126] is applied, both considering fixed rank values (as proposed in [115]) and also using the estimated rank with the proposed approach.

This Chapter is structured as follows. Section 5.2 presents the proposed rank estimation technique. Experimental results are provided in order to empirically show the performance of the rank estimation technique. Section 5.3 introduces the motion segmentation approach used in the experiments. Furthermore, a comparison of results obtained fixing the rank of the matrix of trajectories with missing data or estimating its value with the proposed approach is provided. Section 5.4 gives some examples of the recovered structure and motion of the previously obtained clusters. In addition, a study of robustness of the finally recovered structure and motion for the different objects with respect to features wrongly classified is provided. Finally, concluding remarks are summarized in Section 5.5.

5.2 Missing Data Matrix Rank Estimation

This Section presents the first stage of the proposed approach. The aim is to estimate the rank of the matrix of trajectories W, which contains missing data. This rank is used to fill in the missing entries in W with an imputation technique. In particular, the Alternation technique is used as imputation technique. Chapter 4 shows that, in the single object case, this technique is generally used to find the factors $M_{2f\times r}$ and $S_{r\times p}$ that minimize $\|W - MS\|_F^2$, where r is the rank of matrix W and $\|\cdot\|_F$ is the Frobenius matrix norm [43]. Hence, by multiplying the resulting matrices M and S, a filled matrix, which is the best r-rank approximation to W, is obtained. In the case of a single rigid moving object, and under affine camera model, the rank of W is at

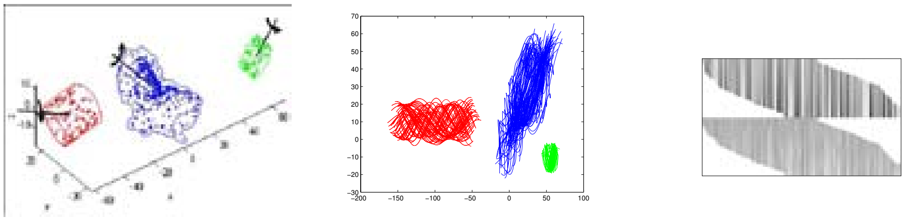


Figure 5.1: (left) scene defined by three synthetic objects (two cylinders and a Beethoven sculptured surface); (middle) feature point trajectories plotted in the image plane; (right) trajectory matrix with 30% of missing data ($W_{30\%missing}$).

most 4.

However, when this idea is extended to the multiple objects case, without any prior knowledge of the scene, the rank of the matrix W is not bounded. Furthermore, due to the missing entries, it is not easy to estimate. The rank of W depends on the number of objects contained in the scene as well as on their motion. Therefore, matrix rank estimation becomes a chicken-egg problem, since the rank value is needed for recovering missing entries and computing an estimation of the full matrix. Moreover, it should be noticed that although several ranks could be tested for filling in missing entries, there is not a direct way to measure the *goodness* of recovered data in order to define which is the most appropriated rank value. Next Section presents a brief study of results that would be obtained if only known entries in W were used for computing its rank.

5.2.1 Partial Information versus Full Information

The importance of using some kind of information about missing entries to estimate the rank of W is presented. This study is performed by using a data set corresponding to a synthetic scene, defined by three rigid objects moving independently. Feature points are distributed over the surface of the objects and tracked through several frames. A full matrix W_{full} is generated and then a trajectory matrix with a 30% of missing data ($W_{30\%missing}$) is obtained by removing information, simulating the behavior of tracked feature points (more information about experiments set up is given in Section 5.2.3). Although this study is presented by using a single case, several tests were considered obtaining similar conclusions (i.e., different percentages of missing data and trajectory matrices defined by the motion of several objects).

Figure 5.1 (left) shows the synthetic scene used through this Section; feature point trajectories, in the image plane, are presented in Fig. 5.1 (middle). Finally, a trajectory matrix with a 30% of missing data is depicted in Fig. 5.1 (right). The half-top of this matrix corresponds to x coordinates of feature points; while half-bottom to y coordinates. White elements correspond to empty entries in the matrix; on the contrary black (or grey) elements in the matrix represent known entries in the matrix.

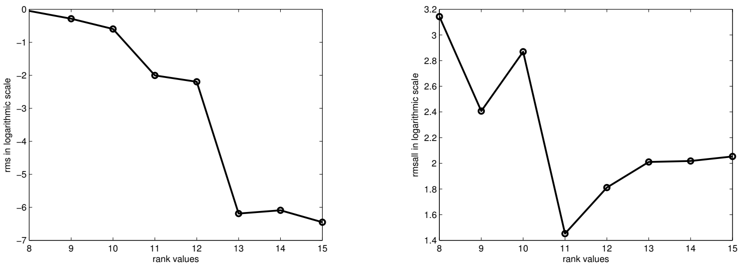


Figure 5.2: (left) rms as a function of the rank values for the missing data matrix presented in Fig. 5.1 (right), when only known entries in $W_{30\%missing}$ are considered; (right) rms_{full} as a function of the rank when all data points in W_{full} are considered.

As mentioned above, the goal at this stage is to fill in empty entries by means of an imputation method assuming different rank values (in our case, the Alternation technique presented in Chapter 3 is used). For every assumed rank value a different solution is obtained (i.e., empty entries in $W_{30\%missing}$ are filled in with different values). In this Section, it is empirically shown that if the considered criterion is based only on known entries, the rank could not be properly estimated. The used criterion consists in selecting the rank value of the matrix with a minimum *root mean square error*(rms). That is, in the current example, the aim is to select the rank for which the matrix obtained with the product $A_{2f\times r}B_{r\times p}$ minimizes the following expression:

$$rms = \|W_{30\%missing} - A_{2f\times r}B_{r\times p}\|_F/\sqrt{q} \tag{5.1}$$

where $2f$ and p are the number of rows and columns in W and q is the number of known entries in $W_{30\%missing}$. On the other hand, since the W_{full} matrix is known, entries filled in during the imputation are also compared with the original values by means of:

$$rms_{full} = \|W_{full} - A_{2f\times r}B_{r\times p}\|_F/\sqrt{q} \tag{5.2}$$

where q is the number of all entries in W_{full}. Hence, two different rms values are computed; the first by only taking into account known entries in $W_{30\%missing}$, while the second by considering all the entries in W_{full}.

The result of this comparison is presented in Fig. 5.2. It can be seen that the trend of these plots is different. Furthermore, the minimum values are obtained at different rank values. In the first case, when only known entries in $W_{30\%missing}$ are used, the best rank estimation corresponds to 15 (Fig. 5.2 (left)); while in the second case, when all entries in W_{full} are used, the best rank estimation is 11 (Fig. 5.2 (right)). Note that the correct rank value is 12, which is quite similar to the value estimated with the rms_{full}; the correct rank value of W_{full} is directly obtained by computing its singular values [43]. Fig. 5.3 shows the 14th first singular values of W_{full}. They are plotted in logarithmic scale in Fig. 5.3 (right). However, in this example, missing entries are filled in better considering $r = 11$ instead of $r = 12$. This can occur due to the random initialization of Alternation.

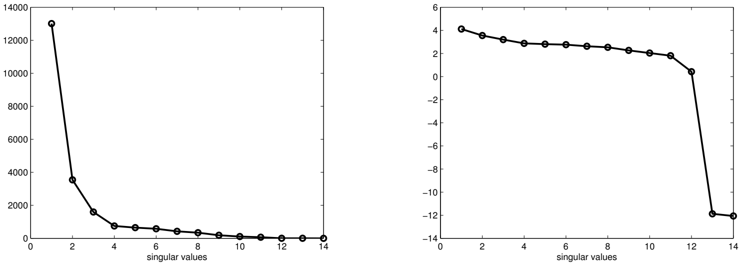


Figure 5.3: (left) first 14th singular values of the matrix W_{full}; (right) the same singular values, but in logarithmic scale.

The previous result can be understood by studying the way missing data in a single trajectory are recovered, after assuming different rank values. Fig. 5.4 presents the trajectory of a given feature point (i.e., a single column in W_{full}), and also the filled in data considering different rank values (concretely, from $r = 8$ up to $r = 14$); thicker line corresponds to known values in $W_{30\%missing}$, thinner line shows original values that were removed to generate the missing data matrix and finally, dashed line corresponds to the filled in entries. An enlargement is presented in the cases of $r = 9, 10, 12, 13$ and 14 in order to obtain a better visualization due to the high values that take the filled in entries in the column (note that in these cases dashed lines go out the range of plot). As illustration, the whole filled in trajectory obtained in the case of $r = 12$ is also plotted. The objective of presenting Fig. 5.4 is to show the way missing entries are filled in by assuming different rank values. It can be appreciated that in all the examples known data are preserved. However, missing data are almost correctly recovered when the assumed rank value is 11, as presented in Fig. 5.2 (right).

5.2.2 Proposed Approach

Having in mind that the goodness of recovered data cannot be measured, since in general there is not a prior knowledge about those missing entries, an approach based on the study of changes on the input matrix after recovering missing data is proposed. Since the rank of W cannot be computed, different rank values (r) are tested, obtaining a full matrix for each case. Then, by using both the initially known and recovered missing entries in W, a novel goodness measurement is introduced. The intuition behind the presented approach is that, since feature point trajectories belong to surfaces of rigid objects, the motion generated by recovered missing entries should be similar to the one of the initial known entries. This motion similarity is identified with the fact that the frequency spectra of the input matrix W should be preserved after recovering missing data. In order to study the frequency content of the matrices, the *Fast Fourier Transform* (FFT) is applied to each of the columns of the full matrices (obtained by using different rank values as presented below) and also to those of the input matrix W (adding zeros to its missing entries). Since the

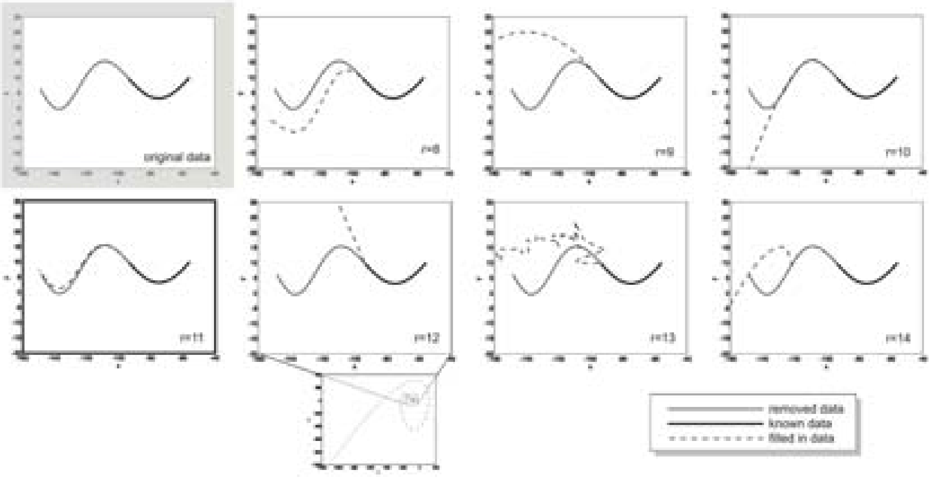


Figure 5.4: (top–left) trajectory of a given feature point (a column in W_{full}); (from top to bottom, from left to right) filled in data (dashed line) assuming rank values from 8 up to 14.

idea is to study the trajectories along the frames, only columns of the matrices are considered, instead of using rows or two dimensions at the same time. Although for a completely different problem, a methodology similar to the one presented in this Chapter has been recently proposed in [34].

Due to the fact that the scene contains more than a single object, an initial rank $r_0 = 5$ is assumed. More elaborated strategies, to find a better estimation of that initial value, could be implemented; for instance by looking for the largest rank of a full submatrix of W. The proposed approach works as follows:

Algorithm: set $r = r_0$ and $k = 0$. First, the modulus of the FFT of the original input matrix W is computed: $F = |FFT(W)|$.

1. By using the current rank r, apply an imputation technique to W to obtain a full matrix of trajectories, W_k (in our experiments the result of Alternation technique, presented in Chapter 3, is used to fill in missing data in W).

2. Apply the FFT to W_k and compute its modulus: $F_k = |FFT(W_k)|$

3. Compute the following difference:

$$e(k) = \|F - F_k\|_F \tag{5.3}$$

4. If $k < 4$, increase both: $r = r + 1$, $k = k + 1$ and go to step 1.

Otherwise, check if $x = k - 3$ is a local minimum of e, that is if x verifies:

$$e(x) < e(x+i), \quad i = 1, 2, 3 \tag{5.4}$$

5. If $e(x)$ is not a local minimum, increase both: $r = r + 1$, $k = k + 1$ and go to step 1.

Solution: The W_k that gives the first local minimum of e is the best full matrix and the corresponding r is the estimation of the rank of W we were looking for.

The three object scene presented in the previous Section is now used to empirically show the performance of the proposed goodness measurement. The same trajectory matrix, with a 30% of missing data, is used ($W_{30\%missing}$). The modulus of the FFT, of the obtained filled matrices considering rank values from 8 up to 14 is plotted in Fig. 5.5. Additionally, the modulus of the original input matrix ($W_{30\%missing}$) is also plotted (Fig. 5.5 (top–left)). It can be appreciated that, the most similar FFT modulus to the one of the initial matrix (Fig. 5.5 (top–left)) is obtained in the case of $r = 11$ (Fig. 5.5 (bottom–left)).

Fig. 5.6 shows the difference between the modulus of $FFT(W_{30\%missing})$ and each one of the FFT_k, obtained considering different rank values. Concretely, the value obtained with equation (5.3) is plotted. It can be seen that the local minimum is found at $r = 11$, which is the same that the one obtained in the previous Section when rms_{all} is considered as a measure of goodness (see Fig. 5.2). Finally, it should be noticed that the shape of the plot presented in Fig. 5.6 is quite similar to the obtained when all data points in W_{full} were considered for plotting the rms_{full} as a function of the rank, compare Fig. 5.6 with Fig. 5.2 (right).

Hence, this measure gives similar results than the one that uses all the entries in the matrix of trajectories. The advantage is that in this case, any prior knowledge about missing data is used.

5.2.3 Experimental Results

A study of the performance of the proposed rank estimation approach is presented in this Section. Synthetic data are used to empirically show the viability of the proposed rank estimation technique. Sequences with different numbers of objects and percentages of missing data (from 10% up to 40%) are considered. For each sequence and each percentage of missing data, 25 attempts are repeated and the obtained rank values are given. These 25 trials are performed to avoid a wrong solution associated with the random nature of the Alternation technique, used to fill in missing data. Recall that it starts with an initial random A_0 or B_0 matrix and proceeds until the product $A_k B_k$ converges to the known values in W.

From a given full matrix, missing data are automatically generated by removing parts of random columns, simulating the behavior of tracked features. The removing

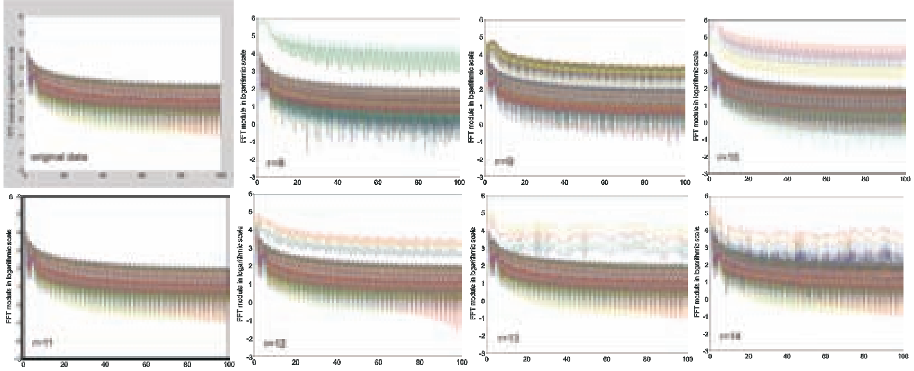


Figure 5.5: modulus of the FFT (logarithmic scale); (top–left) of the original matrix $W_{30\%missing}$; (from top to bottom, from left to right) of the filled matrices considering rank values from 8 up to 14.

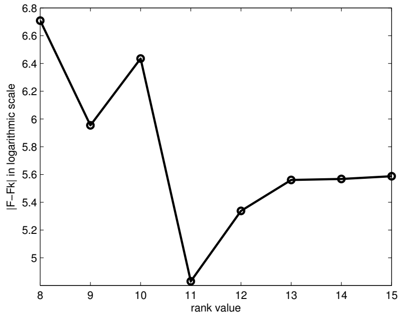


Figure 5.6: difference between F and each one of the F_k (see equation (5.3)); the local minimum is found at $r = 11$.

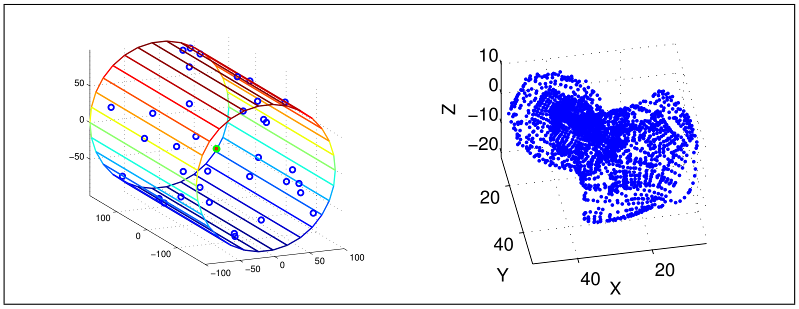


Figure 5.7: synthetic objects; (left) cylinder; (right) Beethoven's sculpture.

process randomly selects a cell in the given column, splitting it up into two parts. One of these parts is randomly removed, simulating features missed by the tracker or new features detected after the first frame, respectively. Since the full matrix is initially known, its rank can be directly computed and compared with the estimated rank obtained with the proposed approach.

Synthetic Data

As in Chapter 4, data sets are generated by randomly distributing 3D feature points over the surface of a cylinder (Fig. 5.7 (left)) and from the triangular mesh (nodes) representing the Beethoven sculptured surface (Fig. 5.7 (right)). Taking these two objects, different sequences are obtained by performing rotations and translations over both of them. At the same time, the camera also rotates and translates. Although self-occlusions are produced, all the points are stacked into the matrix of trajectories, since this is a synthetic experiment. Different number of cylinders and Beethoven sculptured surfaces and different motions are considered in order to generate sequences with more than two objects.

Fig. 5.8 (left) shows the full trajectories obtained from a sequence with two cylinders that move independently. This sequence is defined by 90 frames containing 145 features. The estimated rank values considering different percentages of missing data are plotted in Fig. 5.8 (middle). Details on the polygonal figures can be found in Section 3.3. The rank of the initial full matrix is 8. It can be seen that the rank value is quite well estimated for any percentage of missing data (notice that the obtained median is 8 in all the cases). In particular, the error in the estimation of the rank is less than 20% of the correct value in most cases. The maximum error, which occurs in the case of 40% of missing data, corresponds to an outlier (denoted with a cross in the plot) and the estimated rank value is 50% higher than the correct one. Since the input matrices contain missing data, the *rms* values of filled in matrices are not useful to discern the best rank estimation (see Section 5.2.1); however it gives useful information about how initially known values are preserved. Fig. 5.8 (right) depicts *rms* values corresponding to the examples presented in Fig. 5.8 (middle). It can be appreciated that there is a correlation between estimated rank values and corruption

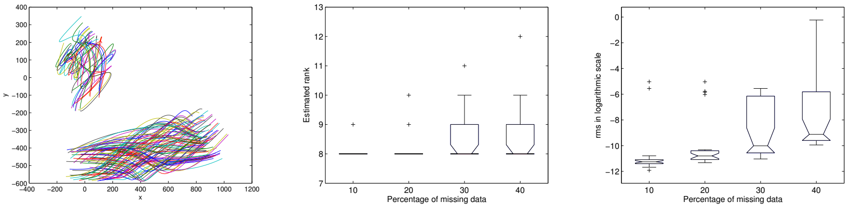


Figure 5.8: two cylinders, rank of full matrix 8; (left) feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

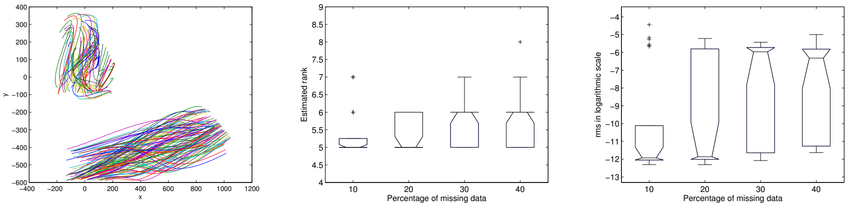


Figure 5.9: two cylinders, rank of full matrix 5; (left) feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

of initially known data. For instance, in the cases of 30% and 40% of missing data, it can be seen that the initially known values have been more corrupted during the filling in process (Alternation) than in the cases of less missing data (see the larger values of the *rms*). Notice that in those cases the error in the estimation of the rank is higher.

Analogously, trajectories of a sequence with two cylinders with dependent motions are plotted into the image plane in Fig. 5.9 (left). In this sequence, the rotation of both objects is identical and consequently, the motion is dependent. Actually, the rank of the matrix of trajectories is 5. The number of features is 160 and the number of frames 90. The rank is quite well estimated for any percentage of missing data, as it is shown in Fig. 5.9 (middle). As in the previous sequence, there is a relationship between the corruption in the initially known values (measured with the *rms*, Fig. 5.9 (right)) and the error obtained in estimation of the rank.

The last example of a sequence with two objects contains a cylinder and a Beethoven sculptured surface. The trajectories are plotted in Fig. 5.10 (left). In this sequence, 451 features (185 from the cylinder and 266 from the sculptured surface) are tracked along 50 frames and the rank of the full matrix is 8. The estimated rank is correctly estimated while the percentage of missing data is below 30%, as it can be appreciated in Fig. 5.10 (middle). Notice that the maximum error in the estimation of the rank is achieved in the case of 40% of missing data; the estimated rank takes a value 15% higher than the correct value. Again, there is a correlation between the error obtained in the estimation of the rank and the value of *rms* (Fig. 5.10 (right)).

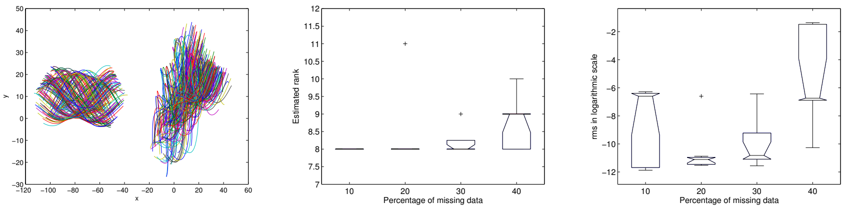


Figure 5.10: a cylinder and a Beethoven sculptured surface, rank of full matrix 8; (left) feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

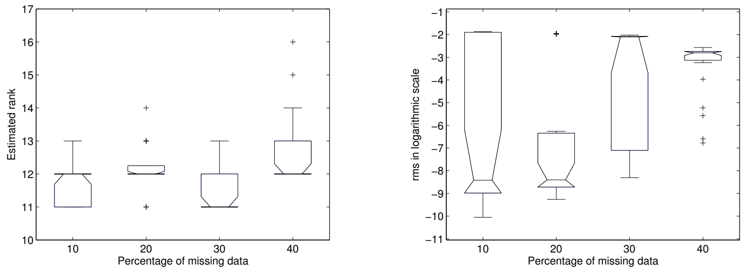


Figure 5.11: three objects, rank of full matrix 12; (left) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

Figure 5.1 (left) shows an illustration of a synthetic scene defined by three objects (two cylinders and a Beethoven sculptured surface). This sequence is defined by 100 frames containing 340 features (114 and 119 from the two cylinders and 107 from the sculpture). Feature point trajectories are depicted in Fig. 5.1 (middle) and a trajectory matrix, for the case of 30% of missing data, is presented in Fig. 5.1 (right). Fig. 5.11 (left) shows the estimated rank values considering the sequence of Fig. 5.1 and different percentages of missing data (the rank of the full matrix is 12). The rank is estimated with an error smaller than 9%, while the percentage of missing data is below or equal to 30%. With a larger percentage of missing data, in some attempts, the error in the estimation of the rank takes values about 16% of the correct rank value. Only in a unique case (which corresponds to an outlier, as it can be seen in Fig. 5.11 (left)) the error takes a value about 33% of the correct value. The *rms* values are plotted in Fig. 5.11 (right). It can be appreciated that, in general, the *rms* takes higher values when the percentage of missing data is about 40%, which corresponds to the case in which the rank is estimated with more error.

Analogously, the obtained results considering sequences of 5, 7 and 9 objects are plotted in Fig. 5.12, Fig. 5.13 and Fig. 5.14. In these sequences, the rank of each input full matrix is 16, 20 and 25, respectively. The corresponding trajectories plotted in the image plane can be seen in Fig. 5.12 (left), Fig. 5.13 (left) and Fig. 5.14 (left).

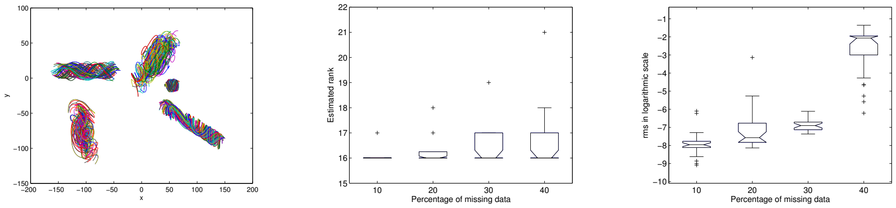


Figure 5.12: five objects, rank of full matrix 16; (left) feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

The estimated rank values are plotted in Fig. 5.12 (middle), Fig. 5.13 (middle) and Fig. 5.14 (middle). In general, the rank is quite well estimated even with a percentage of missing data of about 40%, which is a very remarkable performance considering for instance the large number of objects contained in the last scene—9 objects. Notice that any kind of prior knowledge about the objects contained in the scene nor about their motion is given. Focusing on each sequence, Fig. 5.12 (middle) shows that in more than 75% of cases the rank is estimated with an error smaller than 7% (i.e., about 1) for the case of a sequence with 5 objects. This error is not significant in applications such as motion segmentation.

For the case of the scene with 7 objects (Fig. 5.13 (middle)) the maximum error is reached in the case of 20% of missing data; it is about 20% the correct value. Note that in the rest of cases that error is less than 15% of the correct rank value. Some outliers with a higher error in the estimation of the rank appear in the case of 40% of missing data.

Finally, for the scene with 9 objects (Fig. 5.14 (middle)), the maximum error appears when a percentage of missing data about 40% is considered. That estimated rank value is 28% higher than the correct one. In the rest of cases the rank is estimated with an error smaller than 20% of the correct value. It should be highlighted that the obtained errors are always considerable smaller than if a predefined fixed value were used (e.g., five, as in [117]).

As in the previous sequences, the *rms* value grows as the percentage of missing data increases, as it can be appreciated in Fig. 5.12 (right), Fig. 5.13 (right) and Fig. 5.14 (right) and, again, it is related to the error obtained in the estimation of the rank.

Once the rank is estimated and the missing entries of W are filled in, the motion segmentation of the scenes can be performed (Section 5.3).

Real Data

The same procedure applied to the synthetic data is also used to process real data. The two objects used in Chapter 4 for the real data experiments (see Fig 5.15) are

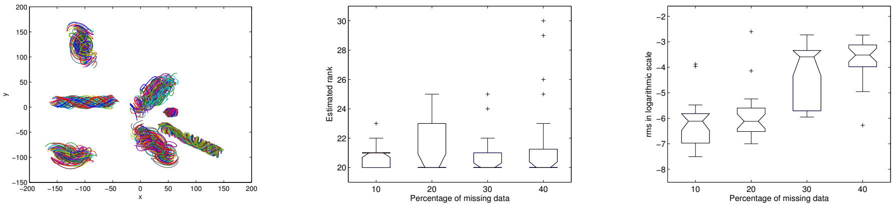


Figure 5.13: seven objects, rank of full matrix 20; (left) feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) rms obtained with Alternation in logarithmic scale.

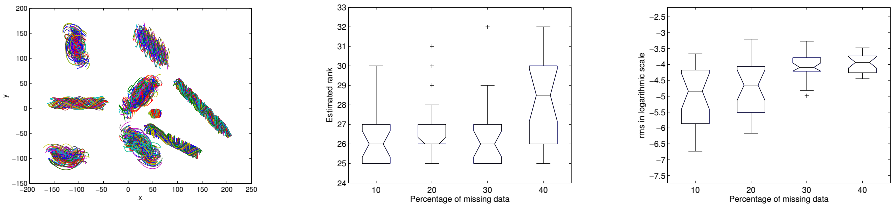


Figure 5.14: nine objects, rank of full matrix 25; (left) feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) rms obtained with Alternation in logarithmic scale.

also considered in this Section.

Full trajectory matrices corresponding to sequences of multiple objects are generated by merging different trajectory matrices of single objects, after swapping x and y coordinates. Overlapping between objects is avoided for the sake of presentation simplicity by applying translations. It should be noticed that, since real images usually contain noisy data, all singular values are nonzero. Therefore, the smallest ones must be truncated in order to estimate the correct rank of W. Since it is difficult to set the appropriated threshold, [71] proposes *model selection* techniques for rank detection. Based on that, the following algorithm is used to estimate the rank of a full matrix in presence of noise:

$$r_m = argmin_r \frac{\lambda_{r+1}^2}{\sum_{j=1}^{r} \lambda_j^2} + \mu r, \tag{5.5}$$

where λ_i corresponds to the i-th singular value of the matrix, and μ is a parameter that depends on the amount of noise. The r that minimizes this expression is considered as the rank of W. The higher the noise level is, the larger μ should be—in our experiments $\mu = 10^{-7}$. This algorithm is used to compute the rank of the full matrices corresponding to real experiments. Hence, its value can be compared to the one estimated with the proposed approach.

The first studied sequence is generated by using the first object twice. The obtained full feature point trajectories are plotted in the image plane in Fig 5.16 (left);

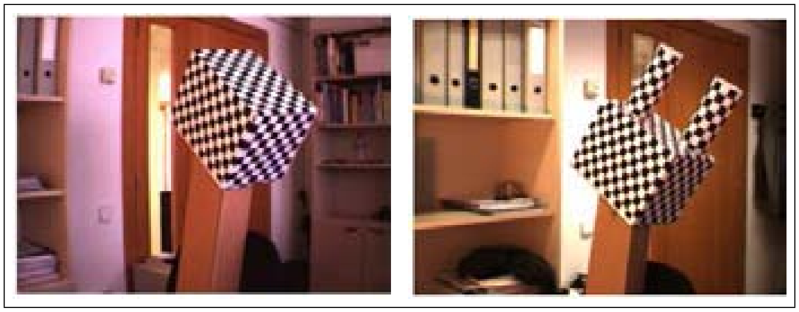


Figure 5.15: real scenes containing two different objects used for the experimental results.

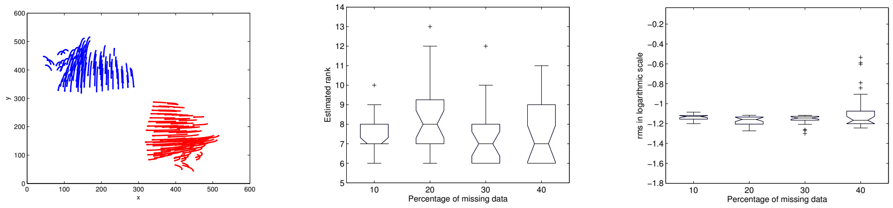


Figure 5.16: first sequence, rank of full matrix 6; (left) full feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

it contains 174 features tracked through 101 frames. A second sequence is generated considering both objects together; the corresponding full trajectories in this second case are plotted in Fig 5.17 (left). It is defined by 61 frames and 275 features (87 from the first object and 188 from the second one). Finally a three object sequence is generated by considering twice the first object together with the second object. This third sequence contains 362 features (87, 87 and 188 from the first and second object respectively), which are tracked through 61 frames. The obtained trajectories are depicted in Fig 5.18 (left).

Fig. 5.16 (middle) shows the estimated rank values for the first sequence. The rank of the full matrix of trajectories is 6; it is obtained by using (5.5) and by setting $\mu = 10^{-7}$. In this particular case, the objects contained in the scene define a degenerate motion. That is, each one of them does not generate a full rank motion matrix (rank 4). In this real data experiment the error in the estimation of the rank is higher than in the synthetic experiments. The median of the estimated rank values takes a value near to the correct rank value in the cases of 10%, 30% and 40% of missing data (concretely, they take a value 16% higher than the correct rank value). The maximum error reach a value of 100% from the correct rank in the case of 20% of missing data.

Similar results are obtained with the second sequence, in which the full matrix has also a rank 6. Fig. 5.17 (middle) shows the estimated rank values. Again, the median

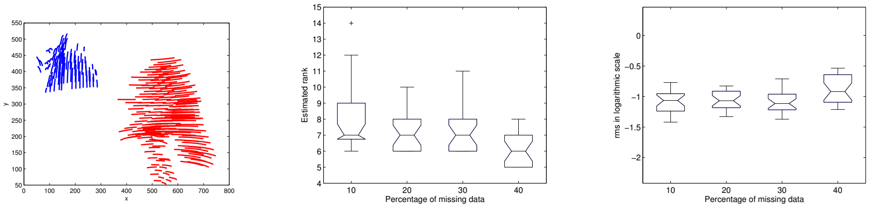


Figure 5.17: second sequence, rank of full matrix 6; (left) full feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

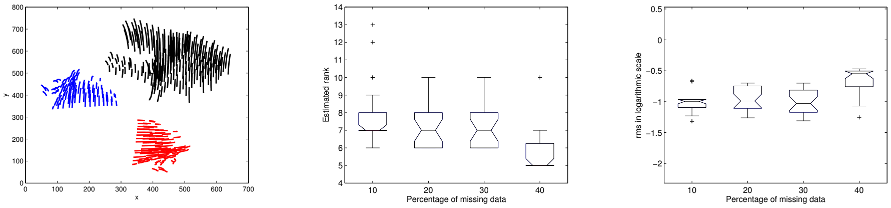


Figure 5.18: third sequence, rank of full matrix 7; (left) full feature point trajectories plotted in the image plane; (middle) estimated rank values for different percentages of missing data; (right) *rms* obtained with Alternation in logarithmic scale.

of the obtained rank values is close to the correct rank value, for any percentage of missing data. The maximum error occurs in the case of 10% of missing data, where the estimated rank value takes a value 100% higher than the correct rank value, in some attempts.

Finally, results for the three object sequence are presented in Fig. 5.18 (middle). In this case, the rank of the full matrix is 7. Although in some cases the error between the estimated rank with the correct ones reaches about ±42%, the median of all the rank values is equal to the correct one, while the percentage of missing data is below 40%.

In the three sequences, the *rms* values, computed with the values initially known in each case, are quite large (see Fig. 5.16 (right), 5.17 (right), 5.18 (right)), in comparison with the synthetic scenes, since the real images contain noisy data. Furthermore, although in general the *rms* values grow as the percentage of missing data increases, the increase in these real data experiments is not as significant as in the synthetic case (see for instance Fig. 5.8 or Fig. 5.12). Hence, it is more difficult to see a correlation between the estimated rank and the *rms*, which takes similar values for any percentage of missing data.

5.3 Multibody Motion Segmentation

This Section presents the second stage of the proposed approach. The aim is to obtain the motion segmentation of the feature point trajectories, once the rank of the matrix of trajectories has been estimated and the missing data in W have been filled in. An algorithm similar to the one proposed by Yan and Pollefeys [126] is used. This algorithm is referred to as Local Subspace Affinity (LSA) in [115].

First of all, the LSA algorithm is briefly introduced. Then a comparison considering different rank values for filling in missing entries in W is presented. The latter is motivated to show the significance of a good rank estimation for correct motion segmentation results. Although preliminary results have been obtained by using a LSA algorithm implemented in the framework of this thesis, finally the implementation publicly available at http://www.vision.jhu.edu/db/ has been used in the experiments to avoid deviation of the results due to a wrong implementation.

Hereinafter, the filled in matrix to which the LSA is applied is denoted as $\mathbf{W}$ to distinguish it from the initial missing data matrix W. The missing data in W has been filled in during the first stage of the proposed approach or with the Alternation by setting a rank value beforehand.

5.3.1 Local Subspace Affinity (LSA)

The approach proposed in [126] consists in estimating a local subspace for each feature trajectory of the filled matrix $\mathbf{W}$, and then compute an affinity matrix based on principal angles [43] between each pair of these estimated subspaces. Finally, the segmentation of the feature trajectories is obtained by applying spectral clustering [122] to this affinity matrix, using the *normalized cut criterion* [103]. The steps of the algorithm are briefly described below.

1. **Rank estimation:** in the first step of the algorithm, Yan and Pollefeys propose to estimate the rank of a full data matrix by using *model selection* techniques [72]. Concretely, they use the expression (5.5) presented above to estimate the rank.

2. **Data transformation:** if $\mathbf{W}$ is a $2f \times p$ matrix, the idea is to consider each of its p columns as a vector in $\mathbb{R}^{2f}$ and to project them onto the unit sphere in $\mathbb{R}^r$, being r the estimated rank value in the previous step.

   The SVD decomposes the matrix of trajectories as $\mathbf{W} = U_{2f\times 2f}\Sigma_{2f\times p}V^t_{p\times p}$. In order to project the trajectories onto $\mathbb{R}^r$, only the first r rows of V^t are considered: $V^t_{r\times p}$. Finally, the p columns of this matrix are normalized to project them onto the unit sphere. As pointed out in [126], this data transformation provides a dimension reduction, a normalization of the data and a preparation for the local subspace estimation in the next step.

3. **Subspace estimation:** for each point α in the transformed space, its local subspace is computed, formed by itself and its n closest neighbours: $[\alpha, \alpha_1, \cdots, \alpha_n]$,

being $n+1 \geq d$; where d is the highest dimension of the linear subspaces generated by each cluster. It is proposed [126] to set $d = 7$ in the articulated motion and $d = 4$ in the rigid object case.

Since the transformed trajectories are defined on the unit sphere, the closest neighbours are selected by using the Euclidean distance.

4. **Affinity matrix:** instead of computing a distance between points, the similarity of each pair of the previously estimated local subspaces is computed and stacked into the affinity matrix $A_{p\times p}$, where p is the number of local subspaces. The affinity of two points α and β is defined as the similarity of their estimated local subspaces $S(\alpha)$ and $S(\beta)$:

$$A(\alpha, \beta) = e^{-\sum_{i=1}^{M} sin(\theta_i)^2}, \tag{5.6}$$

where θ_i is the i-th principal angle [43] between the subspaces $S(\alpha)$ and $S(\beta)$ and M the minimum of their dimensions.

From this definition it can be deduced that the entries of the affinity matrix take positive values, with a maximum of 1. The closer to 1 is an entry $A(\alpha, \beta)$, the more similar are the local subspaces $S(\alpha)$ and $S(\beta)$.

5. **Spectral clustering:** finally, the motion segmentation is obtained by applying spectral clustering [122] to the affinity matrix computed in the previous step. Concretely, the *normalized cut criterion*, presented by Shi and Malik [103], is used to segment the data. Their approach is related to the graph theoretic formulation of grouping, as they pointed out in [103]. Focusing on the problem of motion segmentation, each of the previously computed local subspaces can be interpreted as the nodes of a graph G and the affinity A defines the weights on the edges of the graph. That is, the weight between the node i and j is defined by $A(i, j)$. Let D be the *degree matrix* [122] of A:

$$D(i, i) = \sum_j A(i, j) \tag{5.7}$$

where $i, j = 1 \cdots p$.

The generalized eigenvector y_i is defined as a solution of:

$$(D - A)y_i = \lambda_i D y_i \tag{5.8}$$

Shi and Malik [103] suggest thresholding the second generalized eigenvector of A in order to cut an image into two parts, where the second generalized eigenvector is the y_i corresponding to the second smallest λ_i.

The authors show in [103] that the second generalized eigenvector of A is a solution to a continuous version of a discrete problem in which the goal is to minimize the following expression:

$$\frac{y^t(D - A)y}{y^t D y} \tag{5.9}$$

subject to some constraints omitted for simplicity.

Furthermore, the solution to this discrete problem gives the segmentation that minimizes the *normalized cut*, defined as:

$$Ncut(U, V) = \frac{cut(U, V)}{asso(U, G)} + \frac{cut(U, V)}{asso(V, G)} \tag{5.10}$$

where $cut(U, V) = \sum_{i \in U, j \in V} A(i, j)$ and $asso(U, G) = \sum_{j \in G} \sum_{i \in U} A(i, j)$, and analogously for $asso(V, G)$. More details can be found in [103] and [122].

This criterion (referred to as *Ncut* criterion) measures both the total similarity between the different clusters ($cut(U, V)$) as well as the total similarity within the clusters ($asso(U, G)$ and $asso(V, G)$). The aim is to minimize the similarity between the different clusters $cut(U, V)$, maximizing, at the same time, the similarity within the clusters. This is achieved by minimizing the expression (5.10).

With the $Ncut$ criterion, the data (the local subspaces) are split up into two sets. Working with more than two objects, the idea is to apply this criterion recursively. One of the main drawback of this criterion is that in general it gives over-clustering, since the number of clusters is not known a priori. Actually, Shi and Malik [103] propose a recursive two-way $Ncut$ algorithm. They define a threshold which measures the *stability* of the cut. Hence, a current partition should be subdivided only if the *stability* of the cut is below the defined threshold.

The main problem with the LSA is that many parameters have to be set: the dimension of the projected space (in other words, the rank of W), the dimension of the local subspaces (d) and the number of neighbours that generate the local subspace (n). In [126], the authors propose to estimate the rank of W and the dimension of the local subspaces d by using model selection techniques; the number of neighbours n must accomplish the following relation: $n + 1 \geq d$.

In the current work, the dimension of the projected space r is estimated with the rank estimation approach proposed in the previous Section, d is set to be 4 (since only rigid objects are considered) and n is 6, while $r > 6$ and $r - 2$, otherwise. In addition to the definition of these parameters, it is quite difficult to set a threshold when trying to use the *stability* measure criterion proposed in [103]. The $Ncut$ takes different values, depending on the percentage of missing data in the matrix of trajectories. Finally, a recursive $Ncut$ algorithm that uses the number of objects that are in the scene is implemented. The idea is to segment the data into two clusters recursively until the number of objects is achieved. Although it is not realistic to make use of the number of objects, it is useful to study the performance of the rest of the algorithm.

In order to avoid using model selection techniques, Tron and Vidal [115] propose to fix the rank value (concretely, in their experiments, $r = 5$ and $r = 4N$, where N is the number of different objects in the scene). In the implementation provided at http://www.vision.jhu.edu/db/ , the dimension of local subspaces d is 4 and the number of neighbours is fixed to be 6. Furthermore, the number of objects that contains the scene is also used in the implementation of a recursive $Ncut$ algorithm.

5.3.2 Experimental Results

This Section contains experimental results that aim at comparing results obtained with LSA [126], when the rank of the matrix is estimated with the proposed technique, instead of fixed beforehand.

As mentioned in the introduction, Tron and Vidal [115] propose to fix the rank of W to 5 and to $4N$ when the LSA is applied; the first value is the dimension used in other motion segmentation approach [117] and the second depends on the number of objects in the scene (N). One disadvantage of fixing the rank of the matrix of trajectories to $4N$ is that it assumes both the number of objects contained in the scene (N) and also that all motions are independent and non-degenerate.

In the current experiments, 3 variants are compared: LSA ($r = 5$), LSA ($r = 4N$) and LSA estimating the rank with the previously proposed approach ($r = r_{FFT}$), which does not use any prior knowledge about the scene (such as number of objects or kind of motion). Hence, the Alternation technique is applied considering $r = 5$ and $r = 4N$ in order to fill in the missing entries in W. The results obtained applying the motion segmentation approach [126] to the corresponding filled in matrices are compared with the results obtained by applying the LSA to the filled matrix with the proposed rank estimation technique in the previous stage. As mentioned above, the LSA implementation provided at http://www.vision.jhu.edu/db/ is used in the experiments.

Considering different percentages of missing data, 50 attempts are repeated, because the Alternation is randomly initialized. The percentage of bad-clustered features over the total of features in W is computed in each attempt. This gives a measure of error in the clustering.

Firstly, experiments with synthetic data are carried out. Sequences with 2, 3, 5, 7 and 9 objects generated in the previous Section are used. Secondly, for the real data experiments, two of the sequences with real data presented in the previous Section are studied. Furthermore, sequences from the benchmark presented in [115] are also tested. It consists of a database of 155 video sequences of indoor and outdoor scenes containing two or three motions. These sequences contain degenerate and non-degenerate motions, independent and partially dependent motions, articulated motions and nonrigid motions. The tracked feature point trajectories are full. Based on the content of the video and the type of motion, the sequences can be categorized into three main groups: checkerboard sequences, traffic sequences and articulated/nonrigid sequences. A total of 78 sequences from the first two groups are selected for the experiments: 52 of the checkerboard sequences and 26 of the traffic sequences. Articulated and nonrigid sequences are not studied in this thesis.

As in the previous Section, given a full matrix, the missing data are generated by automatically removing parts of random columns in order to simulate the behaviour of tracked features. Non-filled columns correspond to features missed by the tracker or to new features detected after the first frame (see Section 4.4 for details).

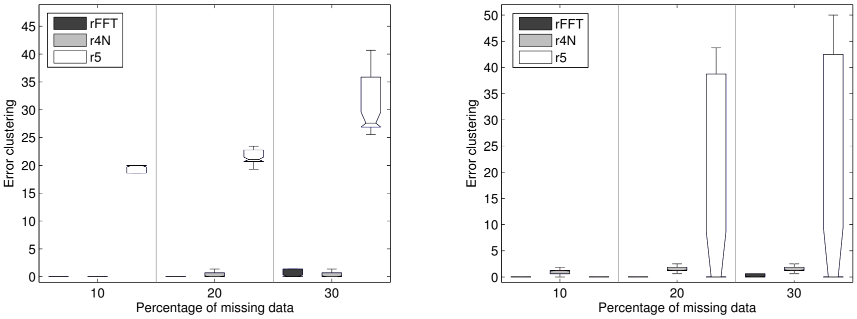


Figure 5.19: 2 cylinders; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$; (left) independent motion; (right) dependent motion.

Synthetic Data

The motion segmentation of the sequences with synthetic data generated in the previous Section are computed. Fig. 5.19 (left) shows the error obtained in the clustering in the case of a sequence where two cylinders move independently. As mentioned in the previous Section, the rank of the initially full matrix is 8. It can be seen that the obtained error when the rank is estimated with the proposed technique is similar to the one obtained when it is fixed to $4N$, where N is the number of different objects in the scene. Notice that in this case, $4N = 8$, which is the rank of W (and the one estimated with the proposed rank estimation approach). Therefore, similar results are obtained in both cases. If the rank is fixed to 5, the obtained error in the clustering is very high.

Fig. 5.19 (right) shows the results obtained in a sequence where two cylinders have dependent motions. As mentioned in Section 5.2.3, the rotation of both objects is identical. In this case, the rank of the initially full matrix is 5. The error obtained when the rank is the estimated with the proposed approach is similar to the error obtained in the previous sequence. In this case, the error obtained assuming $r = 4N$ is higher than considering $r = r_{FFT}$ ($4N = 8$, which is higher than 5). If the rank is set to 5, the obtained error is close to zero, while the percentage of missing data is 10%. With more percentage of missing data, although the median of the error is close to zero, the error takes large values in some attempts.

Analogously, with a sequence that contains a cylinder and a Beethoven sculptured surface (the rank of the initially full matrix is also 8). Fig. 5.20 shows the computed error in the clustering. Again, since $4N = 8$, the error setting $r = 4N$ is similar to the one obtained considering the estimated rank value.

The obtained error in the clustering in a sequence that contains 3 objects is shown in Fig. 5.21. In this case, the rank of the matrix of trajectories is 12. It can be seen that the error obtained when the rank is estimated with the proposed technique, is similar to the one obtained when its value is set beforehand to $r = 4N$ ($4N = 12$, in

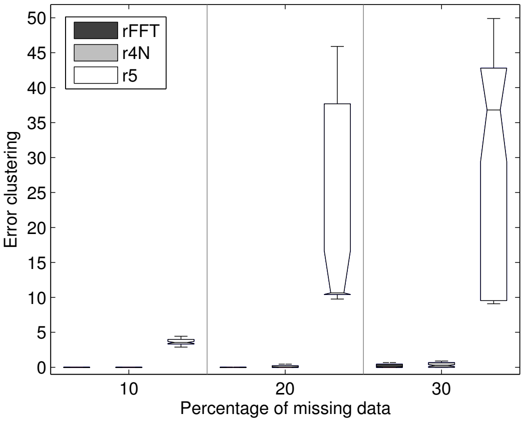


Figure 5.20: a cylinder and a Beethoven sculpture's surface; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

this case). The error obtained in the case of $r = 5$ is very high.

For the cases of 5 and 7 objects, plotted in 5.22 and 5.23 respectively, the error obtained in the clustering is in general smaller when the rank is estimated with the proposed technique. However, the error values are quite similar to the one obtained when the rank is fixed to $4N$.

Finally, Fig. 5.24 shows the error obtained in the clustering when a sequence with 9 objects is considered. It can be seen that similar results are obtained when the rank is estimated with the proposed technique and when it is fixed to $4N$.

Notice that the error obtained when the rank is set to 5 is very high in sequences with more than two objects. As mentioned in the previous Section, the rank of the full matrix in the cases of 3, 5, 7 and 9 objects are 12, 16, 20 and 25, respectively (values very far from 5).

Real Data

The same procedure applied to the synthetic data is now used with real data. As mentioned above, the motion segmentation of two of the sequences generated in the previous Section are computed. Furthermore, sequences from the benchmark presented in [115] are also tested.

Fig. 5.25 shows the error obtained in the clustering in the first sequence, considering different percentages of missing data. It can be seen that the smallest error is achieved considering $r = 4N$, where N is the number of objects in the scene (2 in this case). However, similar results are obtained when the rank is estimated with the proposed technique ($r = r_{FFT}$) and without using any prior knowledge (such as

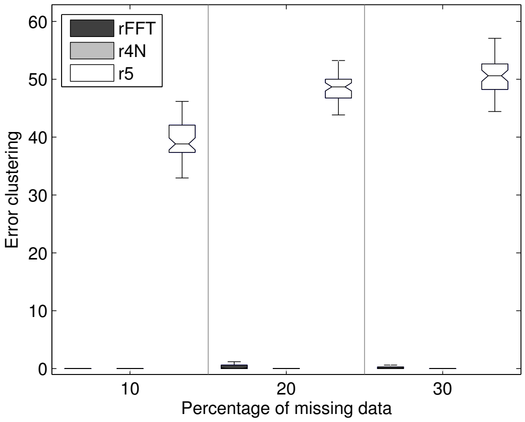


Figure 5.21: 3 objects; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

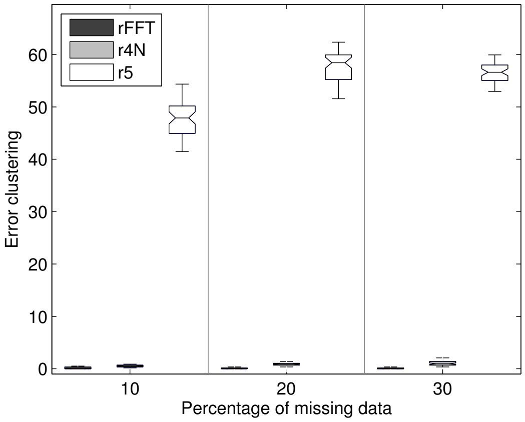


Figure 5.22: 5 objects; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

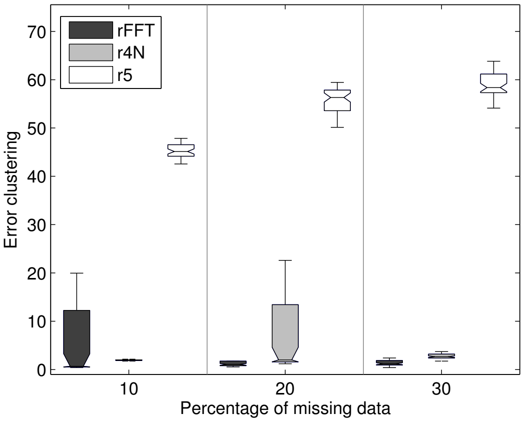


Figure 5.23: 7 objects; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

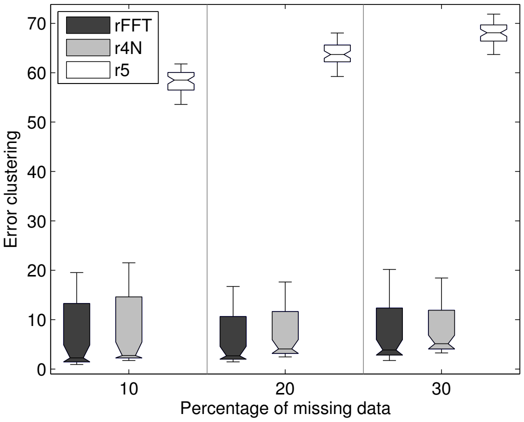


Figure 5.24: 9 objects; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

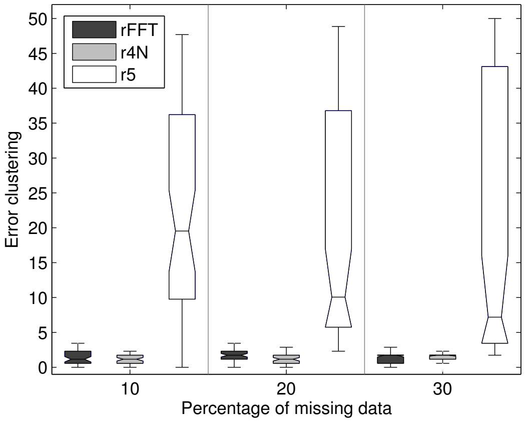


Figure 5.25: first sequence; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

number of objects). The error obtained considering $r = 5$ is very high.

The results obtained with the second sequence are plotted in Fig. 5.26. In this case, similar results are obtained when the rank is estimated with the proposed approach ($r = r_{FFT}$) and with $r = 4N$. Again, the error is very high when the rank is set to 5.

Finally, sequences from the benchmark presented in [115] are considered. As mentioned above, these sequences can be separated into two main groups: checkerboard and traffic sequences. As example, Fig. 5.27 (left) shows the first frame of two sequences from the two groups. The obtained trajectories in each case are plotted in the image plane (Fig. 5.27 (right)).

Fig. 5.28 shows the error obtained in the clustering in the case of the checkerboard sequences. Concretely, Fig. 5.28 (left) and Fig. 5.28 (right) show the obtained error considering sequences with 2 and 3 objects, respectively. It can be seen that, in general, the error in the clustering is smaller when the rank is estimated with the proposed approach ($r = r_{FFT}$). Only in the case of 10% of missing data and for sequences containing 3 objects, the median of the error obtained when $r = 4N$ is smaller than when the rank is estimated with the proposed technique. If the rank is set to $r = 5$, the error is very high, for any percentage of missing data. The obtained error is higher in the sequences with 3 objects, for any rank value and for a percentage of missing data higher than 10%.

Results obtained with traffic sequences are shown in Fig. 5.29. It can be seen that in the case of sequences with 2 objects (Fig. 5.29 (left)), the error obtained in the clustering is smaller when the rank is estimated with the proposed technique, while the percentage of missing data is higher than 10%. If the percentage of missing data is 10%, the smallest error is obtained with $r = 4N$, although similar results are obtained with $r = r_{FFT}$. Furthermore, any prior knowledge about the scene is

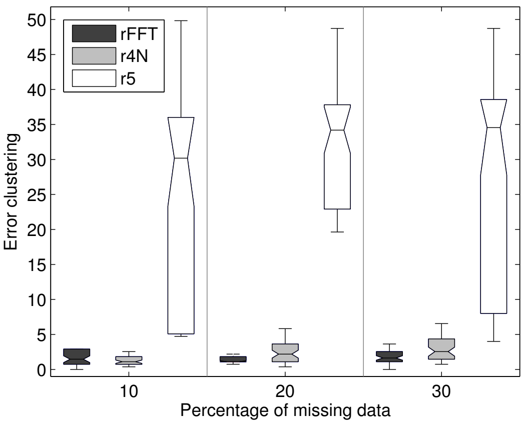


Figure 5.26: second sequence; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$.

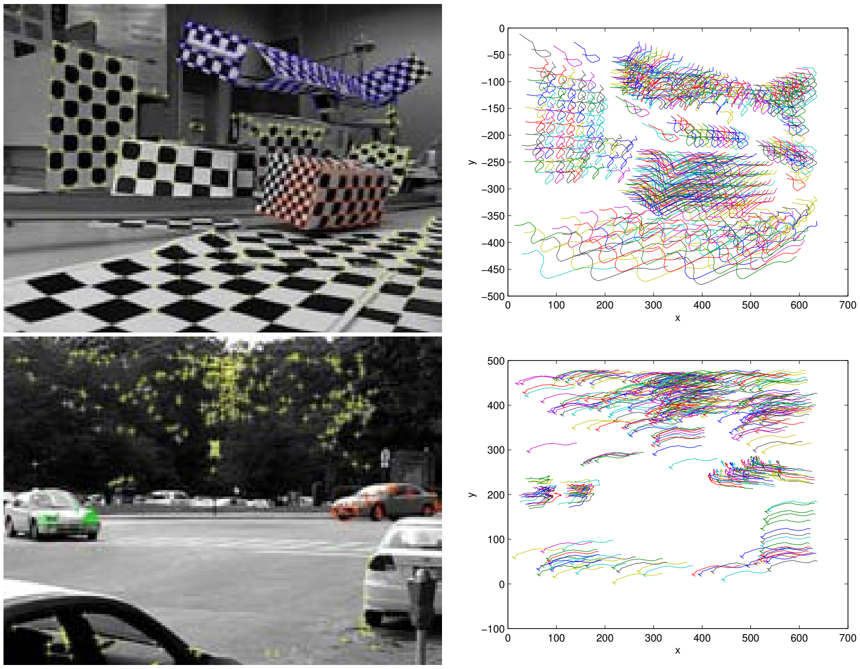


Figure 5.27: (left) first frame of two sequences (feature points are marked); (rigth) trajectories of tracked feature points plotted in the image plane ; (top) checkerboard sequence example; (bottom) traffic sequence example.

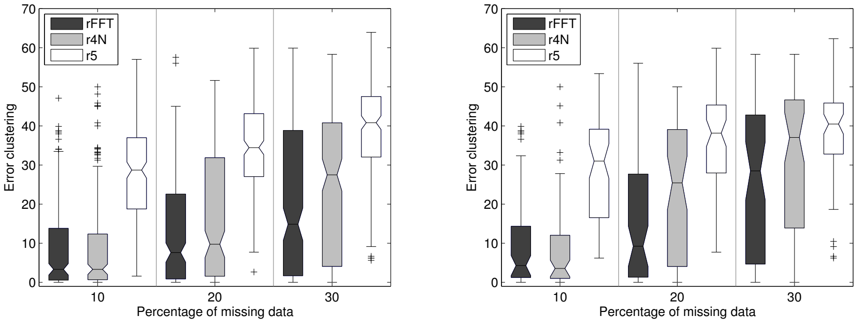


Figure 5.28: checkerboard sequences; error obtained in the clustering considering different percentages of missing data and different rank values: the estimated rank $r = r_{FFT}$, $r = 4N$ and $r = 5$; (left) sequences containing 2 objects; (right) sequences containing 3 objects.

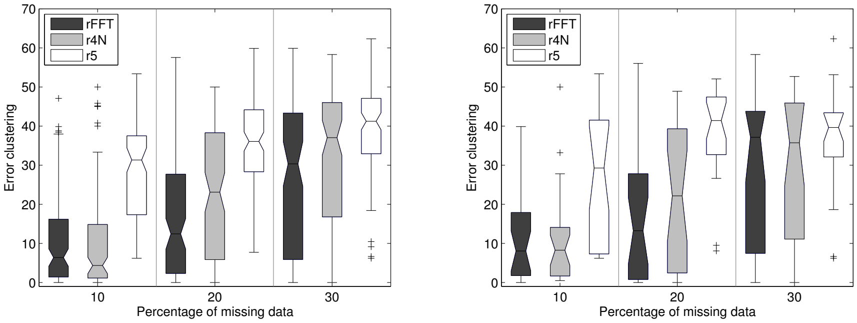


Figure 5.29: traffic sequences; obtained error in the clustering considering different percentages of missing data and different rank values: the estimated rank r_{FFT}, $r = 4N$ and $r = 5$; (left) sequences containing 2 objects; (right) sequences containing 3 objects.

used with the proposed rank estimation technique. In the case of sequences with 3 objects (Fig. 5.29 (right)), the smallest error is obtained when $r = r_{FFT}$, while the percentage of missing data is below 30%. When the percentage of missing data is 30%, the smallest error is obtained with $r = 4N$, although similar results are obtained considering other rank values.

5.4 Structure and Motion of Each of the Clusters

This Section presents examples of recovered structure and motion of the previously obtained clusters. The idea is to apply a single object SFM technique to the trajectories corresponding to each of the clusters. Concretely, the shape and motion of each object are recovered by applying the iterative multiresolution scheme, presented in

Chapter 4, to the initially missing data matrix, considering the columns corresponding to each cluster.

The experiments are focused on the case of 2 objects. However, analogous experiments could be carried out in the case of 3 or more objects. The results would be similar, but with more clusters to study. The goal at this Section is to study the robustness of the recovered factors to wrongly classified features. Although many approaches have been proposed for the motion segmentation problem, as far as we are concerned, none of them present a similar study.

5.4.1 Experimental Results

Synthetic Data

As mentioned above, the shape and motion of each obtained cluster is computed. In the synthetic data experiments, three different plots are presented in each example: the ground truth shape and motion factors, the recovered ones and the overlapping between them. In the case of the motion, three different plots are provided. On the one hand, the evolution of the x and y vectors of the camera along the sequences are plotted separately. Notice that the x and y vectors at frame i, correspond to the unitary camera axes $\mathbf{i}_i$ and $\mathbf{j}_i$ (see formulation in Section 2.1.1). On the other hand, the 2D relative camera-object translation, which correspond to the last column of the motion matrix (see Section 2.1.1), is also plotted. In order to be able to compare the recovered factors and ground truth ones, a process similar the the one proposed in Section 4.4.1 is used. Hence, the ground truth factors are expressed in the same reference system as the recovered ones.

Fig. 5.30 and Fig 5.31 show the recovered shape and motion of a cylinder and Beethoven sculptured surface contained in a sequence presented in Section 5.2.3. This example corresponds to a case with 20% of missing data. The error obtained in the clustering is 0. That is, feature point trajectories are correctly segmented into the different objects. It can be seen that, for both clusters, the shape and motion, plotted in Fig. 5.30 (middle) and Fig 5.31 (middle), respectively, are correctly recovered. Notice that the recovered and the ground truth factors overlap exactly (see Fig. 5.30 (bottom) and Fig 5.31 (bottom)). Actually, the rms_S (eq. 4.17) and rms_M (eq. 4.18) are zero.

In a second example, the sequence used in the previous experiment is considered, but a Gaussian noise with a standard deviation $\sigma = 1$ is added to the coordinates of the feature point trajectories. Again, the percentage of missing data is 20%. Fig. 5.32 (left) and Fig. 5.32 (right) depict the obtained clusters. It can be appreciated that 7 features are wrongly segmented in the first cluster; in fact, the error obtained in the clustering is of about 2.76. In the second cluster, the number of wrongly classified features are 2. Due to those errors and also due to the noise in the data, the shape and the motion are not exactly recovered as it can be appreciated in Fig. 5.33 (middle) and Fig. 5.34 (middle). Notice that the wrongly segmented features (7 in the case of the cylinder and 2 in the case of the Beethoven sculptured surface)

3D shape

relative camera-object motion

x camera vector y camera vector translation

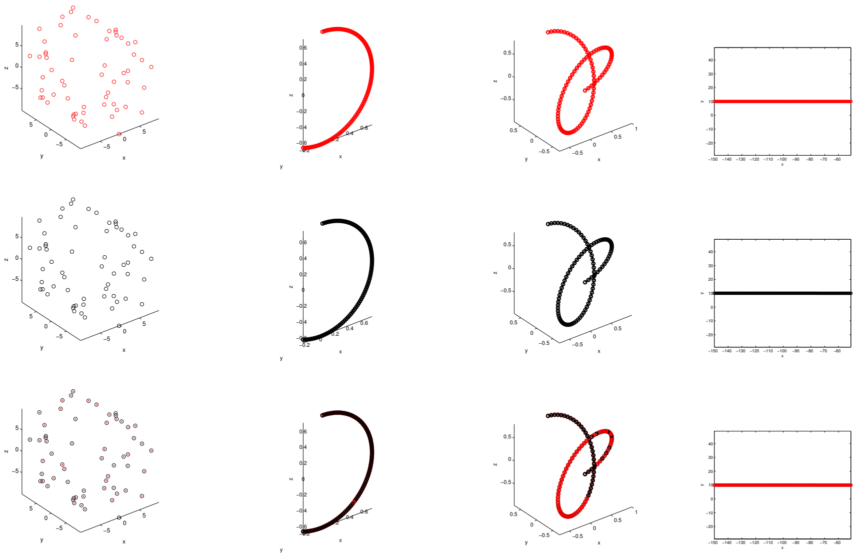
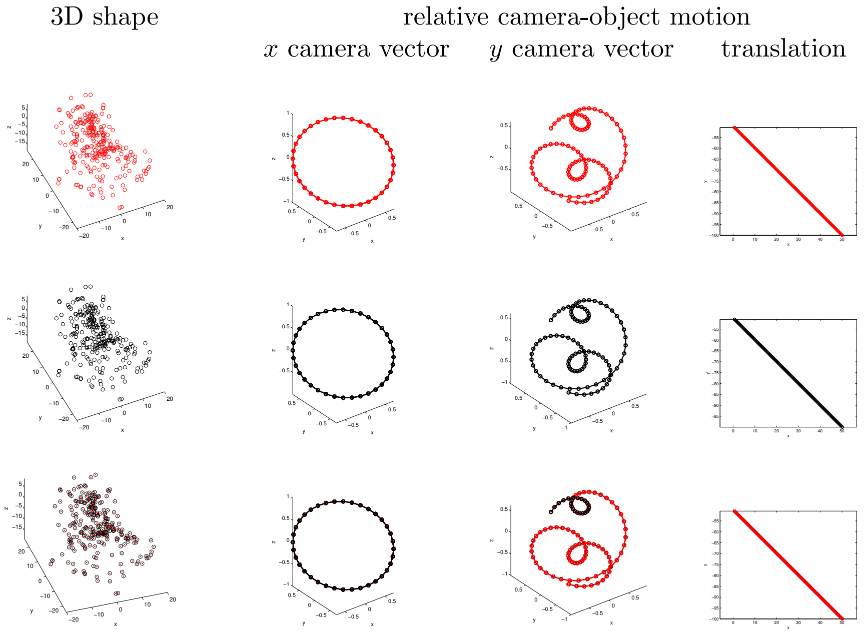


Figure 5.30: cluster corresponding to the cylinder; (top) ground truth factors: 3D shape and relative camera-object motion (3 different plots are provided: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation); (middle) recovered factors; (bottom) ground truth and recovered factors overlapped.


3D shape

relative camera-object motion

x camera vector

y camera vector

translation

Figure 5.31: cluster corresponding to the Beethoven sculptured surface; (top) ground truth factors: 3D shape and relative camera-object motion (3 different plots are provided: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation); (middle) recovered factors; (bottom) ground truth and recovered factors overlapped.

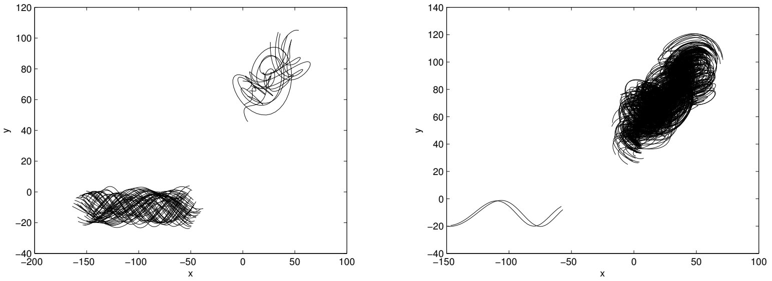


Figure 5.32: clusters obtained in the case of 20% of missing data and $\sigma = 1$; (left) cylinder cluster; (right) Beethoven sculptured surface cluster.

are recovered very far from the others. For both clusters, the ground truth factors and the recovered ones do not overlap (see Fig. 5.33 (bottom) and Fig. 5.34 (bottom)). In this example, the obtained rms_S and rms_M are 1.33 and 0.28 for the first cluster and 0.47 and 0.18 for the second cluster.

Real Data

In the real data experiments, only the recovered factors are shown, since the ground truth ones are not known. The sequence used in these experiments corresponds to one of the sequences of the benchmark presented in [115]. In particular, two objects are considered: the first one is defined by the red feature points over the tubular object on the bottom-left of Fig. 5.35 (left) and the second one is defined by the background feature points (yellow feature points). The first object rotates on one axis and translates on the plane orthogonal, very near to the camera and the camera rotates around its z axis. The sequence consists of 378 points tracked over 29 frames and the considered percentage of missing data is about 10%. The first frame of the sequence is shown in Fig. 5.35 (left) and the feature point trajectories are plotted in Fig. 5.35 (right). In order to obtain the shape and motion of each cluster, the iterative multiresolution scheme presented in Chapter 4 could be applied to the matrix of trajectories corresponding to each obtained cluster. However, with only 29 frames, only a few partitions can be done in the matrix (see Section 4.3 for details) and results would be similar to the ones obtained with the Alternation. Therefore, this technique is used in the experiments with this sequence.

In a first experiment, the error obtained in the clustering is 0. The recovered shape and motion for each cluster are plotted in Fig. 5.36 and Fig. 5.37, respectively. Although in the real case there is no ground truth, it can be seen in Fig. 5.36 (top-right) and Fig. 5.37 (top-right) that the 3D shape of both objects is quite correctly recovered. The recovered motion is plotted in Fig. 5.36 (bottom) and Fig. 5.37 (bottom), for each cluster. Notice that, as in the experiments with synthetic data, three plots are provided to show the recovered camera-object motion: the first and second plots correspond to the x and y camera axes respectively, while the third one corresponds

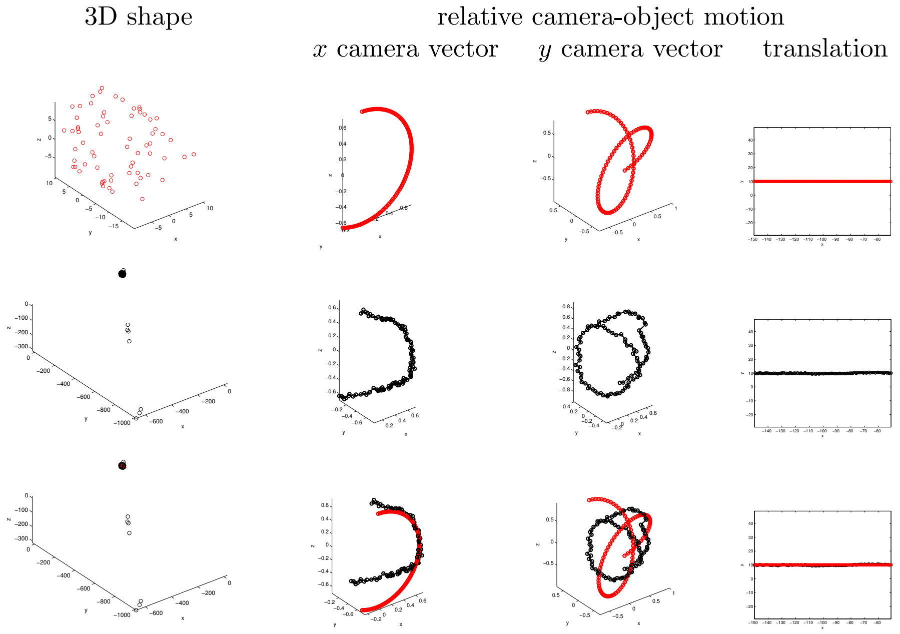
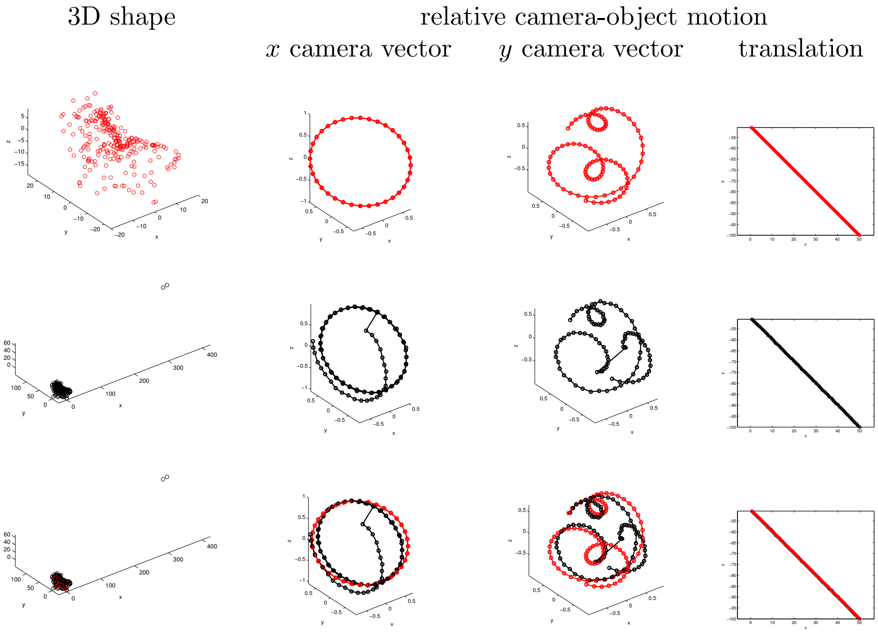


Figure 5.33: cluster corresponding to the cylinder; (top) ground truth factors: 3D shape and relative camera-object motion (3 different plots are provided: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation); (middle) recovered factors; (bottom) ground truth and recovered factors overlapped.



3D shape

relative camera-object motion

x camera vector

y camera vector

translation

Figure 5.34: cluster corresponding to the Beethoven sculptured surface; (top) ground truth factors: 3D shape and relative camera-object motion (3 different plots are provided: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation); (middle) recovered factors; (bottom) ground truth and recovered factors overlapped.

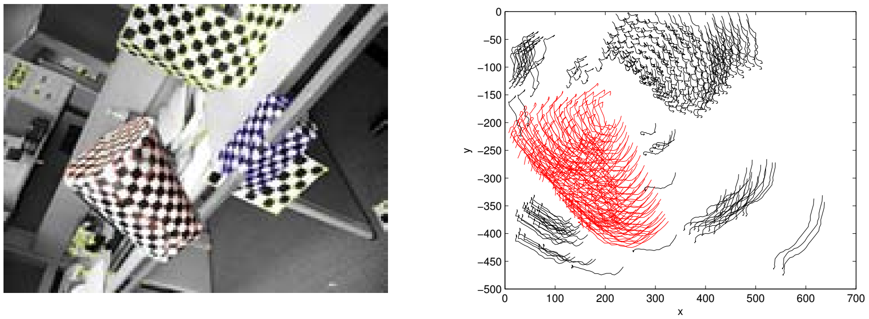


Figure 5.35: (left) first frame of the studied sequence; (right) feature point trajectories plotted into the image plane.

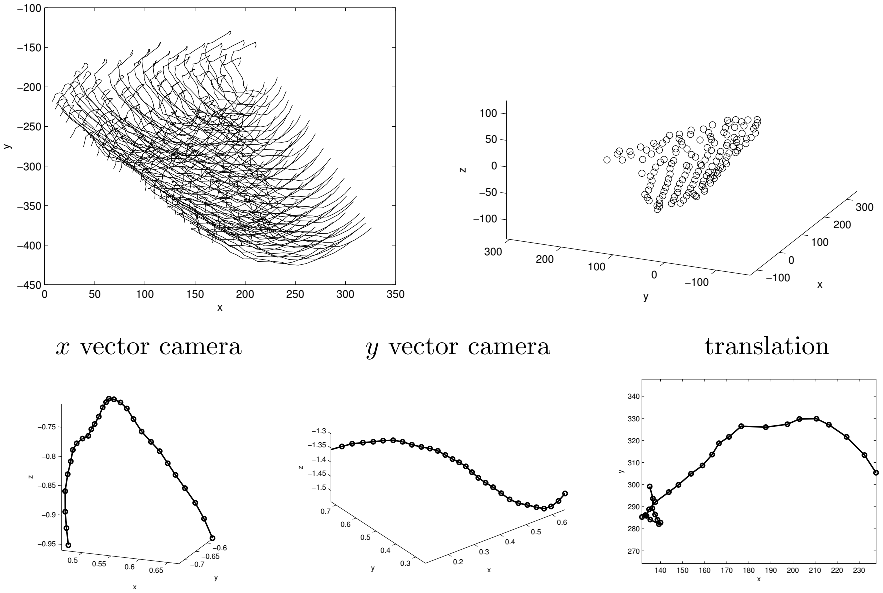
to the translation component.

In a second experiment with real data, the same sequence is considered, but in this case the error obtained in the clustering is about 0.57. The trajectories belonging to each of the clusters are plotted separately in Fig. 5.38 (top-left) and Fig. 5.39 (top-left). In particular, two feature points that belong to the second object are clustered erroneously with the first cluster (see the two trajectories separately from the others in Fig. 5.38 (top-left)). The recovered factors for each obtained cluster are plotted in Fig. 5.38 and Fig. 5.39, respectively. Notice that the 3D shape of the first cluster is not correctly recovered (Fig. 5.38 (top-right)). However, the motion is quite well recovered, as it can be seen in Fig. 5.38 (bottom). On the other hand, the 3D shape of the second cluster is correctly recovered (Fig. 5.39 (top-right)) and also the motion of the camera, as it can be appreciated in Fig. 5.39 (bottom).

5.5 Summary

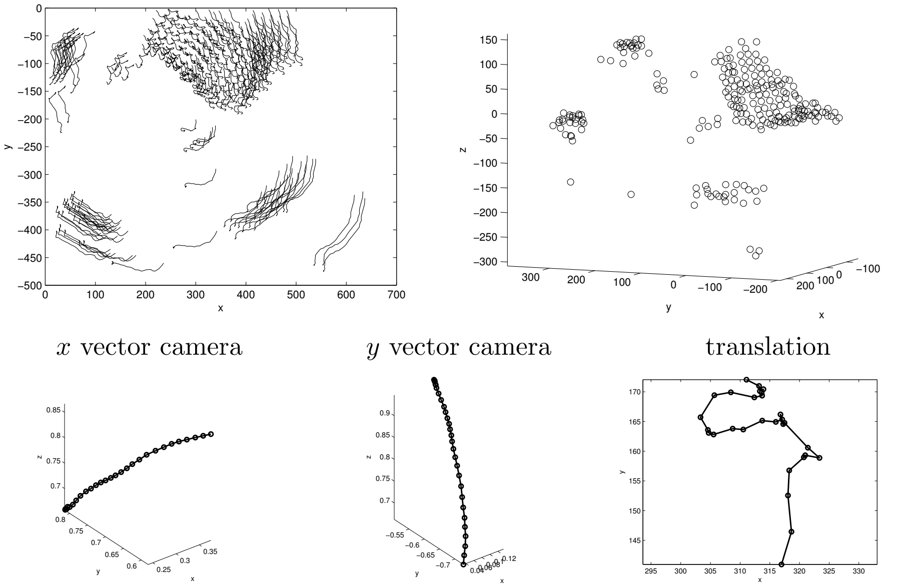
This Chapter presents an approach to tackle the motion segmentation problem with missing data in the matrix of trajectories. It consists of two stages.

In the first stage, a novel technique to estimate the rank of a given trajectory matrix with missing data is proposed. It is based on the study of the frequency spectra of the input matrix W. The intuition behind the proposed approach is that, since feature points belong to surfaces of rigid objects, the frequency content of W should be preserved after filling in missing entries. In other words, the recovered filled matrix should contain a frequency spectra similar to the one of the input matrix. The good performance of the proposed approach is empirically shown, considering scenes with different numbers of objects and different percentages of missing data. From the synthetic scene experiments (no noisy data) it can be concluded that in general, the rank of the input matrix is well estimated, even with percentages of missing data of about 40%. The estimated rank is used to fill in missing data in W by applying the Alternation technique. The rms obtained with Alternation is also analyzed in



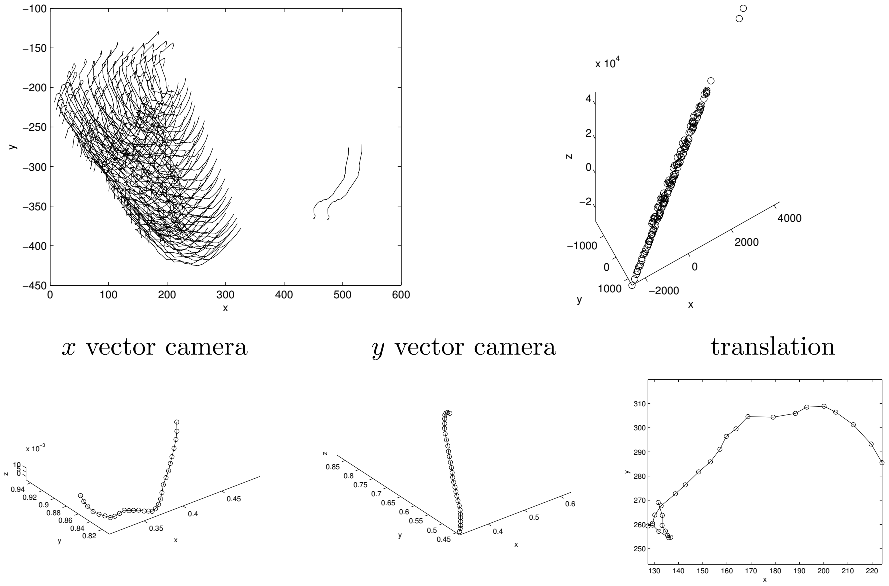
x vector camera   *y* vector camera   translation

Figure 5.36: (top-left) feature point trajectories corresponding to the first object (red feature points in Fig. 5.35 (left)), plotted into the image plane; (top-right) recovered 3D shape; (bottom) recovered motion: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation.



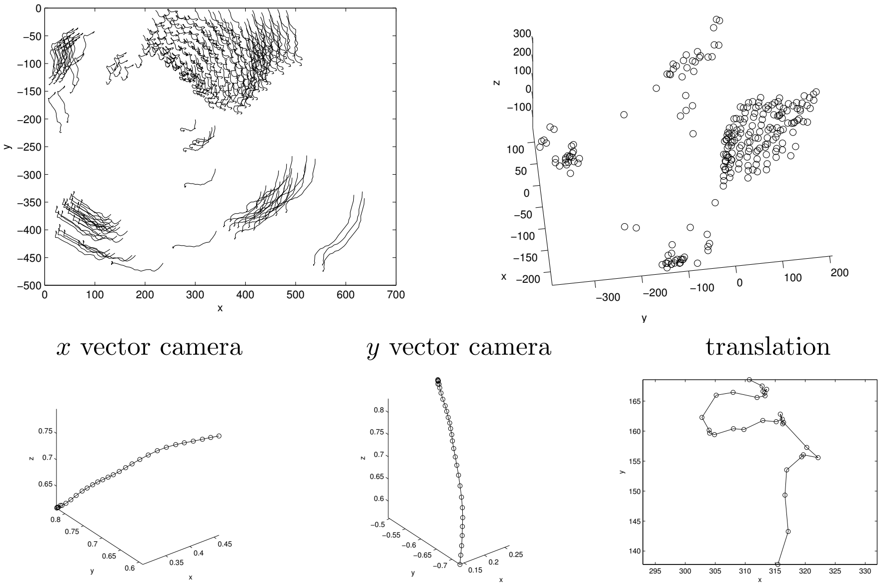
x vector camera y vector camera translation

Figure 5.37: (top-left) feature point trajectories corresponding to the second object (yellow feature points in Fig. 5.35 (left)), plotted into the image plane; (top-right) recovered 3D shape; (bottom) recovered motion: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation.



x vector camera y vector camera translation

Figure 5.38: (top-left) feature point trajectories corresponding to the first cluster, plotted into the image plane; (top-right) recovered 3D shape; (bottom) recovered motion: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation.



x vector camera

y vector camera

translation

Figure 5.39: (top-left) feature point trajectories corresponding to the second cluster, plotted into the image plane; (top-right) recovered 3D shape; (bottom) recovered motion: evolution of the x and y vectors of the camera along the sequence and 2D relative camera-object translation.

order to study the error added to the data during the filling in process. Real data experiments show that the proposed technique is also able to deal with noisy images. It should be highlighted that the proposed rank estimation approach does not require the prior knowledge of the objects contained in the scene, nor assumption about their motion.

In the second stage, given a full matrix of trajectories, an approach similar to the one proposed in [126] is used to obtain the motion segmentation. In order to show the significance of a good rank estimation for correct motion segmentation results, a comparison considering different rank values for filling in the missing data matrix is performed. It has been shown that the obtained error in the clustering is in general smaller when the rank is estimated with the rank estimation technique proposed in the previous Section, instead of fixing it beforehand. In some cases, the error is equal or a little higher than when the rank is fixed to be $4N$, but with the proposed approach, no prior information about the scene (such as number of objects or kind of motion) is used.

Finally, the shape and motion of every obtained cluster are computed, both for cases where the error obtained in the clustering is null and for cases where there are wrongly segmented features. The experiments show that the correctness of the recovered factors depends on the correctness of the obtained clusters. In the examples, with only a few wrongly segmented features, the shape and the motion are not correctly recovered.

Chapter 6

Applications

The aim at this Chapter is to show that the Alternation technique can be used in other applications apart from the Structure from Motion (SFM) problem. Concretely, three different applications are considered. An introduction is provided for every proposed application, in order to set the problem to tackle. Then, the Alternation is adapted to the particular application and finally, experimental results that include real data are given.

6.1 Introduction

As mentioned in Chapter 1, several problems can be reduced to find a low-rank matrix approximation of the data matrix: sequences alignment [129], faces decomposition [121], optical flow estimation [65], motion segmentation [126], photometric stereo [56] and data mining [116]. In the SFM problem, for instance, it is shown in Chapter 4 that the trajectory matrix can be approximated by a matrix of rank 3 or 4 (or even higher), depending on the hypothesis of the problem. The SVD can be used to find this low-rank matrix approximation when the data matrix is full, but SVD method can not be used with missing data matrices. In those cases, the Alternation gives the best low-rank approximation matrix in the least squares sense. In some cases, the final goal is to compute the factors, while in others the objective is to fill in missing entries in the given matrix with the product of those factors. The aim at this Chapter is to apply the approaches proposed along the thesis to other problems, different from the SFM. First of all, different problems in which data can be stored into a matrix and that can present missing data are selected. Then, each problem is studied, searching for similarities and differences with the SFM problem. Finally, the Alternation is adapted to tackle each of the studied problems.

The first application proposed in this Chapter consists in applying the Alternation technique to tackle the photometric stereo problem. The goal is to recover the surface normals and reflectance of an object by using intensity images obtained under different lighting conditions. Furthermore, the initial images can be approximated by multiplying these obtained factors. The grey intensities of the images are stacked into a *measurement matrix*. Concretely, each row of this matrix corresponds to a given pixel in the different images, while each column corresponds to a single image. Hence, data matrices have in general a large number of rows. This matrix can be factorized into the surface matrix (which contains the normals and reflectance of the surface) and the lighting source matrix (which contains the light source directions and intensities). These recovered factors allow to represent the object in any lighting condition, which can be very useful in object detection or recognition tasks, for instance. The key point is that the set of images of a Lambertian object can be approximated by a low-dimensional linear subspace. In particular, if the images do not have shadows nor specularities, the *measurement matrix* can be approximated by a rank-3 matrix. Otherwise, a larger rank is needed. In fact, when working with real images, shadows and specularities are quite common. Furthermore, there is usually an ambient background illumination that affect image intensities. In order to tackle these situations, a rank-4 formulation, which includes an ambient illumination term, is introduced. Entries that correspond to specularities or to pixels in shadow are considered as missing data. Hence, they do not affect the final result. Missing data do not follow a random distribution, since they are generated by considering upper and lower thresholds of intensity values.

In a different framework, an Alternation-based strategy is applied in order to fill in missing entries in a matrix of gene expression data. The data matrix contains the *log*-transformed ratios of expression levels of genes under different experimental conditions. These ratios are computed with the information provided by *DNA microarrays*. This technology allows the whole genome to be monitored on a single chip so that a better picture of the interactions among thousands of genes can be observed simultaneously. Missing values may be due to insufficient resolution, image corruption, dust or scratches on the slide, or simply as a result of the robotic methods used to create them. The data are stacked into a data matrix in which the rows correspond to each of the studied genes, while the columns correspond to each of the experiments. In general, a large amount of genes under a few different experiments are studied. Corrupted entries are manually or automatically considered as missing data, since they may influence wrongly the final results. The structure of the missing data is a combination of random missing and what is called *line-missing* data. This last structure occurs when entries corresponding to a concrete gene are missing nearly in all the experiments. One possible solution to avoid missing data could be to repeat the experiment, but it would be a very expensive task. The main drawback in this application is that the rank of the data matrix is equal to the number of columns. Due to that fact, when working with missing data, results obtained with the Alternation are not good as expected. The proposed Alternation-based strategy consists in applying the Alternation considering $r = 1$, filling in the missing data. Then, the filled matrix is projected onto a $(r + 1)$-dimensional subspace. This process is repeated till r is equal to the number of columns. Thus, both the initially missing and known entries are well recovered. Although the proposed strategy does not always outperforms current state of the art in this field, a novel field where the Alternation can be used is shown.

Finally, the Alternation is adapted to be used in recommender systems. The main objective of recommender systems is to help customers to find products they will probably like or dislike. These systems are widely used on *Electronic commerce*, commonly known as *e-commerce*, which consists in buying and selling products or services over electronic systems such as Internet and other computer networks. In many cases, these systems use collaborative filtering techniques in order to give recommendations to the customers. These techniques use the information given by *similar* customers to recommend or predict rates of the current customer. A similarity measure should be defined in order to find neighbour customers. However, this measure can be not significant when the customers do not have common rated items or have only a few. As other proposed approaches, the Alternation uses all the information stored in the system to find the predictions. The data matrix contains rates that customers (rows) give to some products (columns). In general, there is a wide variety of products and a large number of customers. Consequently, data matrices are very large. In addition, these matrices tends to be very sparse, since each customer rates only a few items. One advantage that should be used in this particular application is that the range of the rates values is known. Therefore, in order to improve the final results, the Alternation is modified by imposing the range of values in the computed predictions (i.e., the range of the filled in entries).

Unfortunately, in these applications, the dimensions of the data matrix make difficult to apply the iterative multiresolution scheme proposed in Chapter 4. Furthermore, only in the last application the data matrix has a high percentage of missing data, which is a necessary condition to obtain remarkable improvements with the proposed iterative scheme.

6.2 Photometric Stereo

The image of an object depends on many factors such as lighting conditions, viewpoint, geometry and albedo of the object, etc. Small changes in lighting conditions can cause large changes in appearance. Hence, if a general description of the object is required, it is necessary to design a representation that capture all the image variation caused by those factors. Then, it could be used for object detection and recognition, for instance.

Photometric stereo aims at estimating the surface normals and reflectance of an object by using several intensity images obtained under different lighting conditions. The general assumptions are that the projection is orthographic, the camera and objects are fixed and the moving light source is distant from the objects. Hence, it can be assumed that the light shines on each point in the scene from the same angle and with the same intensity.

The starting point (as pointed out in [8]) is that the set of images produced by a convex Lambertian object, under arbitrary lighting, can be well approximated by a low-dimensional linear subspace of images. In [102], it is shown that in the absence of shadows, a Lambertian object produces a 3D subspace of images. Basri et al. [7] show that higher dimensional subspaces should be used to account for the effects of attached shadows, which correspond to points in which the angle between the surface normal and the direction of the light source is obtuse (i.e., higher than 90 degrees). This linear property suggests to use factorization techniques to model the image formation and obtain each of the factors that contribute to it. The intensity of the pixels of the image are stacked into a *measurement matrix*, whose rows and columns correspond to each of the pixels and images, respectively. Since the number of pixels is larger than the number of images, the *measurement* matrix has a larger number of rows than columns.

One common assumption in most photometric stereo approaches is that the images do not have shadows nor specularities (e.g., [133]), which correspond to points with very low and high intensities values, respectively. This is due to the fact that these points do not follow a Lambertian model. Although if there are only a few of them the Lambertian model is a good approximation, their presence can bias the obtained results. Hence, some approaches propose methods to reject them or tend to reduce their influence on the results.

Hayakawa [56] presents a photometric stereo approach for estimating the surface normals and reflectance of objects, which is similar to the factorization method presented in [110] for the shape and motion estimation. Assuming that the object's surface is Lambertian, the surface normals, the surface reflectance, the light-source direction and the light-source intensity are determined simultaneously. This approach is based on SVD and, in order to obtain a unique decomposition, it uses one of the following constraints: 1) at least six known pixels have constant or known reflectance; 2) the light source intensity value is constant or known in at least six known images. However, in practice it is needed to know the value of the reflectance in those pixels or the intensity values in those images, respectively. Otherwise, the intensity and reflectance values are recovered only up to scale. Hayakawa proposes a strategy to deal with shadows. First of all, shadows and illuminated data are classified, by using an intensity threshold. The idea is to select an initial submatrix, whose entries do not correspond to pixels in shadow. Then, the surface normals and reflectance of pixels in shadow are estimated by growing a partial solution obtained from the initial submatrix. Unfortunately, to sort the matrix and find a submatrix without shadows is in general a very costly task. Furthermore, the SVD has a high computational cost when dealing with big matrices, which are common in this application.

Epstein et al. [33] present an approach for learning models of the surface geometry and albedo of objects and that is based on Hayakawa's approach [56]. Concretely, they propose four different schemes that are used depending on the information available in each case. The first scheme uses the knowledge of the light source direction. In this case, shadows and specularities are not taken into account in the optimization process. The other schemes are based on the SVD and shadows and specularities can not be removed. Epstein et al. point out that in [56] the obtained reflection and light directions are recovered up to a rotation. In order to solve that ambiguity, they propose the *surface integrability*.

Yuille et al. [128] propose and extension of [33] in which the SVD is used to obtain the shape and albedo of objects under different unknown illumination. In this case, no prior information is used. Furthermore, the authors propose a generalization of the SVD formulation that allows to recover the ambient illumination. It is shown that if several variant light directions are used, specularities do not affect the results. Unfortunately, there is not always a variety of illuminations in the given images. They also propose a method to locate and reject shadows. It consists of an iterative process whose initialization is the result given by the SVD. The idea is to consider shadows as outliers and remove them of the function to minimize.

Basri et al. [8] also use the photometric stereo to recover the shape and reflectance properties of an object. The main advantage of this recent approach is that it allows arbitrary lighting, including any combination of point sources and diffuse lighting. They use spherical harmonics [7], which form an orthonormal basis for describing functions on the surface of a sphere. In particular, Basri et al. present two methods, the first one uses a first order harmonic approximation (a 4D space), while the second one uses a second order harmonic approximation (a 9D space). Although the last gives better results, it involves an iterative optimization and requires at least nine images. They propose to remove unreliable pixels, such that saturated pixels, and to fill in the missing data by using Wiberg's algorithm [123].

This thesis proposes to use the Alternation technique to decompose the *measurement matrix*, which contains the intensity images, into the surface and light source matrices. On the one hand, shadows and specularities are considered as missing data. Thus, the results do not depend on those pixels, which in general influence the final results. On the other hand, a rank-4 formulation that includes an ambient illumination term is introduced.

This Section is structured as follows. First of all, the image generation process in the case of a Lambertian surface is introduced. Then, the formulation used along the Section and the Alternation technique adapted to the photometric stereo are exposed. Experimental results both with synthetic and real images are given. Finally, concluding remarks are summarized.

6.2.1 Lambertian Model

The image intensity at the pixel (x, y) depends on the optical properties of the surface material (albedo), the surface structure and the spectral distribution of the incident illumination. As pointed out in [56], for many surfaces the fraction of incident illumination reflected in a particular direction depends only on the surface normal. The reflectance characteristics of such a surface can be represented as a *reflectance* function ϕ of the three unit vectors: surface normal $\mathbf{n} = (n_x, n_y, n_z)^t$, light-source direction $\mathbf{m} = (m_x, m_y, m_z)^t$, and the viewer direction $\mathbf{v} = (v_x, v_y, v_z)^t$. Fig. 6.1 shows the *geometric reflectance model* for image generation in the viewer-oriented coordinate system. Using the reflectance function ϕ, the following equation describes the image-generation process:

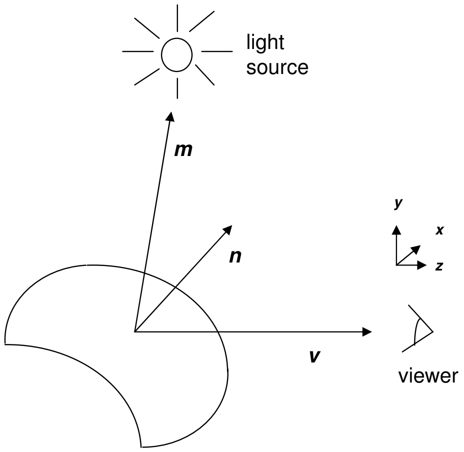


Figure 6.1: geometric reflectance model for image generation in the viewer-oriented coordinate system.

$$i = t\phi(\mathbf{n}, \mathbf{m}, \mathbf{v}) \tag{6.1}$$

where t contain the light source intensity at each image.

Assuming that the image projection is orthographic and that there is only a distant point light source, the viewer direction and the light source direction can be considered to be constant over the image plane.

The most used reflectance model is the *Lambertian* model, which states that materials absorb light and reflect it uniformly in all directions. This model is given by the following equation:

$$I(x, y) = r(x, y)\mathbf{n}(x, y)t\mathbf{m} \tag{6.2}$$

where $r(x, y)$ is the albedo at the (x, y) pixel, $\mathbf{n}(x, y)$ is its surface normal and $\mathbf{m}$ contain the light direction at each image. The albedo at each point on the object $r(x, y)$, describes the fraction of light reflected at that point. Or, which is the same, it describes how much light is reflected by each surface normal. It is important to remark that the amount of light reflected depends on the surface normal at that point, and it does not depend on its spatial position.

6.2.2 Formulation

A *measurement matrix* I contains the grey-level intensity image data at p pixels through f frames in which only the light source is moving. In particular, the kth-row of I corresponds to the intensities of the kth-pixel in each of the images. At the same time, the jth-column of I corresponds to the intensities of all the pixels of the jth-frame. Hence, the matrix I is defined as:

$$I_{p\times f} = \begin{bmatrix} i_{11} & \dots & i_{1f} \\ \vdots & & \vdots \\ i_{p1} & \dots & i_{pf} \end{bmatrix} \tag{6.3}$$

The space of images of the object obtained by varying the light source direction spans a three dimensional space [102], if there are not shadows or specularities. Therefore, it can be assumed that the rank of I is 3. Assuming a Lambertian reflectance model, this matrix can be factorized as:

$$I = RNMT \tag{6.4}$$

where

$$R_{p\times p} = \begin{bmatrix} r_1 & & 0 \\ & \ddots & \\ 0 & & r_p \end{bmatrix} \tag{6.5}$$

is the surface reflectance matrix (being r the surface reflectance at each pixel),

$$N_{p\times 3} = \begin{bmatrix} \mathbf{n}_1 \dots \mathbf{n}_p \end{bmatrix}^t = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ \vdots & \vdots & \vdots \\ n_{px} & n_{py} & n_{pz} \end{bmatrix} \tag{6.6}$$

is the surface matrix ($\mathbf{n}$ represents the surface normal at each pixel),

$$M_{3\times f} = \begin{bmatrix} \mathbf{m}_1 \dots \mathbf{m}_f \end{bmatrix} = \begin{bmatrix} m_{x1} & \dots & m_{xf} \\ m_{y1} & \dots & m_{yf} \\ m_{z1} & \dots & m_{zf} \end{bmatrix} \tag{6.7}$$

is the light-source direction matrix ($\mathbf{m}$ represents the light-source direction at each frame), and

$$T_{f\times f} = \begin{bmatrix} t_1 & & 0 \\ & \ddots & \\ 0 & & t_f \end{bmatrix} \tag{6.8}$$

is the light-source intensity matrix (t represents the light-source intensity at each frame).

Using the above definitions, the surface matrix S and the light-source matrix L are defined as follows:

$$S_{p\times 3} = \begin{bmatrix} \mathbf{s}_1 \dots \mathbf{s}_p \end{bmatrix}^t = \begin{bmatrix} s_{1x} & s_{1y} & s_{1z} \\ \vdots & \vdots & \vdots \\ s_{px} & s_{py} & s_{pz} \end{bmatrix} = RN \tag{6.9}$$

$$L_{3\times f} = \left[\begin{array}{c} \mathbf{l}_1 \ldots \mathbf{l}_f \end{array}\right]^t = \left[\begin{array}{ccc} l_{x1} & \ldots & l_{xf} \\ l_{y1} & \ldots & l_{yf} \\ l_{z1} & \ldots & l_{zf} \end{array}\right] = MT \tag{6.10}$$

Therefore, the *measurement* matrix can be decomposed as:

$$I = SL \tag{6.11}$$

Hence, the surface matrix S and the light-source matrix L can be recovered from the intensity images obtained under varying illumination. Furthermore, and once the factors are obtained, synthetic images can be generated, considering arbitrarily light positions and substituting them to the expression (6.11). Therefore, images obtained under different light illumination can be simulated.

6.2.3 Alternation Adapted to the Photometric Stereo Problem

The problem presented above could be tackled by any factorization technique. In general, Singular Value Decomposition (SVD) is used to compute the S and L factors from a measurement matrix I; however, if entries of I corresponding to pixels in shadow or saturated regions (also denoted as specularities) are considered as missing data, SVD can not be applied. In this section, an adaptation of Alternation [54], which is able to deal with missing data, is proposed to factorize the matrix I. The algorithm is summarized below for the rank 3 case, the extension to the rank 4 case is presented in the next Section.

Algorithm:

1. Set a lower and an upper threshold to define the shadows and specularities, respectively. The lower threshold depends on the intensity values in each set of images, while the upper threshold is, in general, 255.

2. Consider the entries corresponding to shadows and specularities as missing data in I.

3. Apply the Alternation technique to I. The algorithm starts with an initial random $p \times 3$ matrix S_0 (analogously with a $3 \times f$ random L_0) and repeats the next two steps until the product $S_k L_k$ converges to I:

   - Compute L^1:
   $$L_k = (S_{k-1}^t S_{k-1})^{-1}(S_{k-1}^t I) \tag{6.12}$$
   - Compute S^1:
   $$S_k = I L_k^t (L_k L_k^t)^{-1} \tag{6.13}$$

[1] These products are computed only considering known entries in I.

Solution: S contains the surface normals and reflectance, L contains the light source direction and intensities and its product SL is the best rank-3 approximation to I.

However, as in the SVD case, the obtained decomposition is not unique, since any invertible matrix Q with size 3×3 gives the following valid decomposition:

$$I = SL = \hat{S}QQ^t\hat{L} \tag{6.14}$$

Therefore, at the end of the algorithm, one of the constraints proposed in [56] is used to determinate the matrix Q:

1. The relative value of the surface reflectance is constant or known in at least six pixels. The matrix Q can be computed with the following system of p equations:

   $$\hat{s}_k QQ^t \hat{s}_k^t = 1, \quad k = 1, \cdots, p \tag{6.15}$$

   where $\hat{s}_k$ is the kth-vector of $\hat{S}$.

2. The relative value of the light-source intensity is constant or known in at least six frames. Here Q can be obtained by solving the following system:

   $$\hat{l}_k^t QQ^t \hat{l}_k = 1, \quad k = 1, \cdots, f \tag{6.16}$$

   where $\hat{l}_k$ is the kth-vector of $\hat{L}$.

If the value of the reflectance or the value of the light intensity is known, it is substituted to the corresponding above equation. Actually, if the value is not known, the reflectance and the light intensity are recovered only up to scale.

In our experiments, the second constraint is used and a total of f equations (the number of available images) are considered instead of only six.

The recovered normal vectors N and the light source directions M are represented in an arbitrary coordinate system. However, they can be represented in the viewer-oriented coordinate system if at least three surface normals or three light source directions are known. Then, N and M can be aligned to the viewer coordinate system.

6.2.4 Generalization to the Rank 4 Case

This Section proposes a generalization of the previously presented formulation to the rank-4 case. It includes a term corresponding to the ambient background illumination. With this new formulation, the equation (6.11) is transformed as:

$$I_{p\times f} = S_{p\times 3}L_{3\times f} + \mathbf{a}_{p\times 1} \tag{6.17}$$

where $\mathbf{a}_{p\times 1}$ is the ambient illumination, which does not depend on the light source direction. It could take a different value at each pixel.

In matrix formulation, this equation can be expressed as:

$$I_{p\times f} = \left[\begin{array}{cc} S & \mathbf{a} \end{array}\right] \left[\begin{array}{c} L \\ \mathbf{1} \end{array}\right] \tag{6.18}$$

Notice that each of the factors can be of rank 4 at most. Therefore, in this case, the Alternation technique is applied considering $r = 4$ and the following decomposition is obtained:

$$I_{p\times f} = \widetilde{A}_{p\times 4}\widetilde{B}_{4\times f} \tag{6.19}$$

At each step of the Alternation technique, it is imposed that the last row of $\widetilde{B}$ is a vector of ones, in a similar way than in the SFM problem. As in the rank-3 case, some constraints should be imposed in order to obtain a unique decomposition. Actually, the same constraints are imposed, but the formulation is generalized as follows:

$$I = \left[\begin{array}{cc} S & \mathbf{a} \end{array}\right] \left[\begin{array}{c} L \\ \mathbf{1} \end{array}\right] = \tilde{A}_{p\times 4}\tilde{B}_{4\times f} = A_{p\times 4}Q_{4\times 4}Q^t_{4\times 4}B_{4\times f} \tag{6.20}$$

The linear transformation $Q_{4\times 4}$ can be computed using one of the aforementioned constraints, depending on the conditions of the problem. In order to compute it more easily, this matrix is separated into two different matrices: Q_1 and Q_2 with dimensions 4×3 and 4×1, respectively. That is,

$$I = AQQ^tB = A\left[\begin{array}{cc} Q_1 & Q_2 \end{array}\right] \left[\begin{array}{c} Q_1^t \\ Q_2^t \end{array}\right] B = \left[\begin{array}{cc} AQ_1 & AQ_2 \end{array}\right] \left[\begin{array}{c} Q_1^tB \\ Q_2^tB \end{array}\right] \tag{6.21}$$

If the surface reflectance is constant or known in every pixel, the matrix Q is computed by solving the linear systems defined by the equations:

$$a_kQ_1Q_1^ta_k^t = 1, k = 1, \cdots, p \tag{6.22}$$

where a_k is the kth-vector of the first factor A, and

$$\mathbf{a} = AQ_2 \tag{6.23}$$

On the contrary, if the intensity of the light source is constant or known in every image, the matrix Q is computed by solving the linear systems defined by:

$$b_k^tQ_1Q_1^tb_k = 1, k = 1, \cdots, p \tag{6.24}$$

where b_k is the kth-vector of the second factor B, and

$$\mathbf{1} = Q_2^t B \tag{6.25}$$

Finally, the Q matrix is used to obtain the final factors:

$$\tilde{A} = AQ, \quad \tilde{B} = Q^t B \tag{6.26}$$

6.2.5 Experimental Results

The aim at this Section is to show that results are improved when pixels in shadow and specularities are considered as missing entries in I. Hence, results obtained taking the full image intensity matrix are compared with the ones obtained when those particular entries are considered as missing data.

The goodness of the results is studied showing the recovered factors. In the experiments with synthetic data, the recovered factors can be compared with the ground truth ones by computing the rms of each of the factors. The following expressions are used as measures of goodness:

$$rms_R = \frac{\|R_{gt} - R\|_F}{\sqrt{p}} \tag{6.27}$$

$$rms_N = \frac{\|N_{gt} - N\|_F}{\sqrt{3p}} \tag{6.28}$$

$$rms_M = \frac{\|M_{gt} - M\|_F}{\sqrt{3f}} \tag{6.29}$$

$$rms_T = \frac{\|T_{gt} - T\|_F}{\sqrt{f}} \tag{6.30}$$

where the ground truth factors are denoted by the subindex gt.

Unfortunately, in the experiments with real data the ground truth factor is not known in general and it is difficult to compare different results. However, in the current experiments with real data, the light source direction at each frame is known. This information can be used to align L and N with the viewer-coordinate system.

Synthetic Images

In this first experiment, a synthetic sphere is generated. There is not an ambient component nor noise in the data. The light source directions are generated by simulating a trajectory on a sphere and avoiding positions of the light source behind the objects. A total of 66 images with a size of 71×71 pixels are generated. Therefore, the size of the measurement matrix is 5,041×66. Finally, the matrix used into the process has a size of 1,941 × 66, since only the pixels that are not background are used. Several negative intensity images are obtained due to attached shadows. That is, pixels in which the normal and light-source direction vectors form an obtuse angle. Since the intensities can not be negative in real situations, they are set to 0 in the experiment.

Table 6.1 summarize the results obtained with the synthetic images. Concretely, the results obtained when the negative intensities are set to 0 are shown in Table 6.1 (a).

Table 6.1: results case synthetic images.

experiment	rms_R	rms_N	rms_M	rms_T
(a) if $I(x,y) < 0$, $I(x,y) = 0$	0.053	0.17	0.094	1.06
(b) 30% missing and Alternation	3.10×10^{-16}	4.07×10^{-16}	3.59×10^{-16}	9.52×10^{-15}
(c) 61.05% missing and Alternation	88.15	4.84	0.092	2.60
(d) 61.05% missing, iterative multires. scheme and Alt.	0.015	0.043	0.042	0.72

If data in shadow are considered as missing entries, that is, entries corresponding to pixels which have negative intensities, the *measurement* matrix has a percentage of missing data of about 30%. The results obtained in this case are shown in Table 6.1 (b). Notice that the error of all the recovered factors is very close to zero. Fig. 6.2 shows the recovered factors in this case. It can be seen in Fig. 6.2 (top) that the surface normals and reflectance are correctly recovered. This is an ideal situation and the light source position is recovered exactly (see Fig. 6.2 (bottom)). Therefore, the recovered images are identical to the original ones used to estimate the factors, as it can be appreciated in Fig. 6.3.

If the images contain large regions in shadow, the percentage of missing data can be very high and the results obtained with the Alternation can be worse than expected. In those cases, the iterative scheme presented in Chapter 4 could be used to improve final results. As an example, a high percentage of missing data is generated in the current *measurement* matrix, by setting a threshold of 5 (the maximum intensity value in this particular matrix is 10). That is, pixels with an intensity image below 5 are considered as missing data, giving a *measurement* matrix I with a percentage of missing data of 61.05%. The results obtained when the Alternation is applied to this matrix are shown in Table 6.1 (c). It can be seen that the error is higher than the obtained in previous cases. If the iterative scheme presented in Chapter 4 is applied to the matrix I, which has a percentage of known data of only 38.95%, a matrix with a percentage of known data of 84.55% is obtained. The Alternation applied to this filled matrix gives the results shown in Table 6.1 (d). Notice that the error is smaller than in the previous experiment. However, it is higher than the obtained in the second experiment. This is only an example to show that the iterative scheme

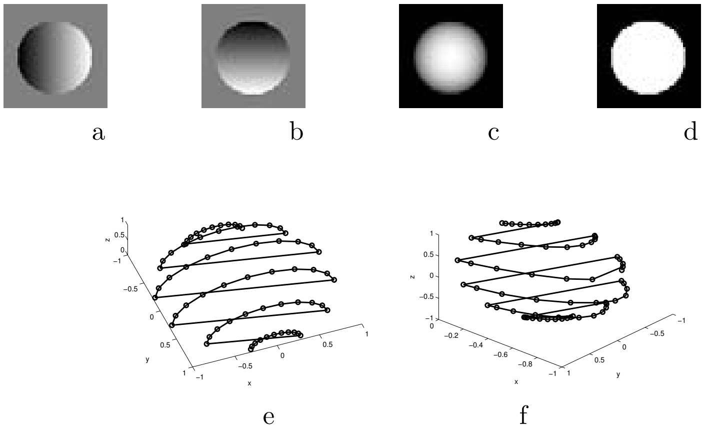


Figure 6.2: synthetic sphere images, pixels in shadow are considered as missing data, given a matrix with 30% of missing data; (top) x,y and z components of the recovered surface normals and recovered reflectance (d); (bottom) ground truth (e) and recovered light position (f).

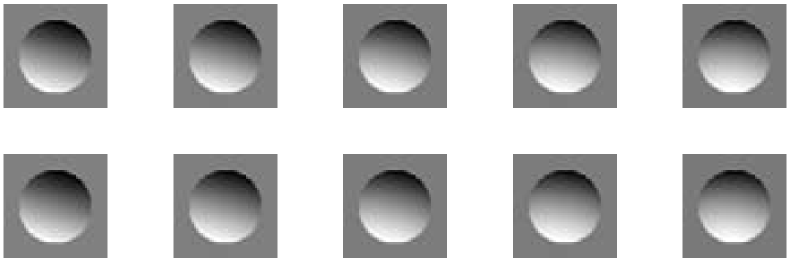


Figure 6.3: (top) a set of the original images synthetically generated; (bottom) images recovered by projecting the original ones onto a three dimensional subspace, considering the pixels in shadow as missing data.

can be used in cases with high percentage of missing data.

POV-Ray Images

This second experiment, looking for more realistic simulation, uses the *POV-Ray software* to generate images that contain a sphere. In this case, there is background ambient illumination in the scene. The light source directions generated in the previous experiment are used. The images are of 120×120 pixels, given rise to a measurement matrix of 14,400 × 66 elements. Since only the entries corresponding to non-background pixels are considered, the final resulting matrix contains 7,499×66 elements.

Fig. 6.4 shows the recovered factors in the case of non-missing data. It can be seen in Fig. 6.4 (a) that the reflectance is constant in all the points of the surface. Fig. 6.4 (c), Fig. 6.4 (d) and Fig. 6.4 (e) show each of the coordinates of the recovered surface normals. Notice that there are points on the boundaries of the visible surface in which the normal values are not properly recovered (see the enlargement provided in the case of the z-coordinate, Fig. 6.4 (e-right)). This is due to the fact that these points correspond to shadow pixels in several images.

Fig. 6.5 provides a comparison of some original images with the recovered ones, when projecting to a rank-4 subspace. The first row in Fig. 6.5 shows five of the initial images. The second row contains the images obtained with the product of the recovered factors S and L, while the third row consists of these images plus the ambient term. Notice that pixels corresponding to regions in shadow are also in shadow in the recovered images.

In order to correct the recovered regions in shadow (see Fig. 6.5 (bottom) and those incorrectly recovered values in the normals, shadows are considered as missing data. Concretely, the pixels whose intensity is lower than 120 are considered as missing data. With such a high threshold, a percentage of missing data of about 57% is obtained. Fig. 6.6 shows the results corresponding to this missing case. Here, the normal components do not present incorrect values in the boundaries.

The recovered images obtained in this case are shown in Fig. 6.7. In this case, the areas corresponding to pixels in shadow in the original images are not as marked as in the previous experiment, as it can be appreciated in the third row in Fig. 6.7. Therefore, the results are improved considering the pixels in shadow as missing data, even with a high percentage of missing data.

The proposed iterative scheme is not used, since the size of the matrix is very high. Therefore, the partitions proposed in the iterative multiresolution scheme to obtain overlapping among the filled in regions could not be achieved here. Actually, only the rows would be partitioned. Furthermore, the method would have a high computational cost due to the large number of rows.

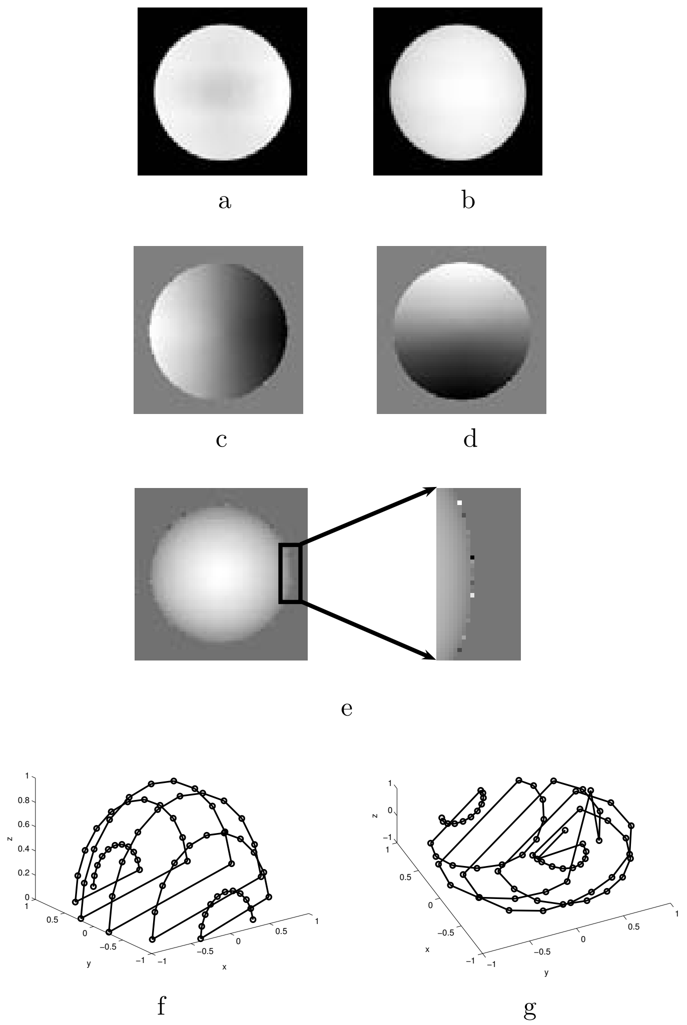


Figure 6.4: sphere images obtained from data generated with *POV-Ray*; (a) recovered reflectance; (b) recovered ambient illumination; (c) x-coordinate of the recovered surface normals; (d) y-coordinate of the recovered surface normals; (e) z-coordinate of the recovered surface normals. An enlargement is presented in order to obtain a better visualization of the points on the boundary; (f) ground truth of the light position; (g) recovered light position.

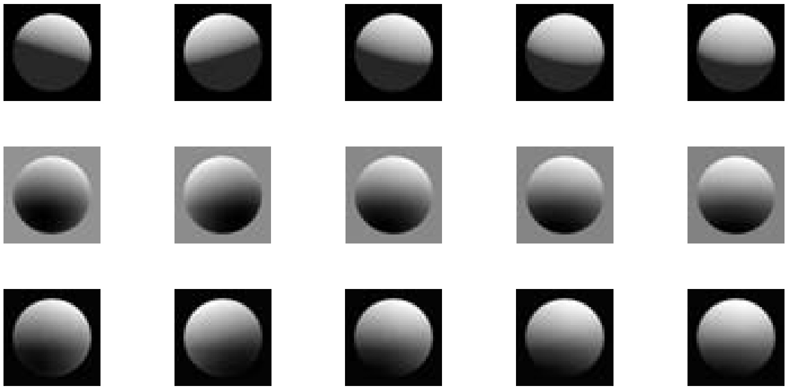


Figure 6.5: (top) a set of the original images generated with the POV-Ray; (middle) images recovered by projecting the original ones onto a four-dimensional subspace; (bottom) images of the middle, adding the ambient component.

Real Images

Images from the Yale data base (http://cvc.yale.edu) are used in the experiments with real data. All data sets have been generated under controlled light. Concretely, images are captured using a purpose-built illumination rig. This rig is fitted with 64 computer controlled strobes. The 64 images of an object in a particular pose are acquired at camera frame rate (30 frames/second) in about 2 seconds. The extreme cases, in which almost all the pixels of the image are in shadow, are not considered in these experiments. At the end, only 49 images are taken to construct the *measurement matrix*. In particular, two different data sets are used; a scene containing: i) a ball; ii) a sculptured bust.

Ball Images

These images contain many regions of specular reflection, that is saturated pixels with an intensity of 255 (see examples in the first row in Fig. 6.9). The images have a size of 294×294 pixels, which give a measurement matrix of 66,921 rows and 49 columns. If the background pixels were considered, the matrix would have 86,436 rows.

Fig. 6.8 shows the results obtained taking all the intensity values. The surface normals of some background pixels, which are not studied, can be appreciated in Fig. 6.8 (c), Fig. 6.8 (d) and Fig. 6.8 (e).

Fig. 6.9 gives a comparison between the original images and the recovered ones with the product of the obtained factors, without the background ambient term (second row) and with the ambient term (third row). It can be seen that the specular regions keep quite specular in the recovered images.

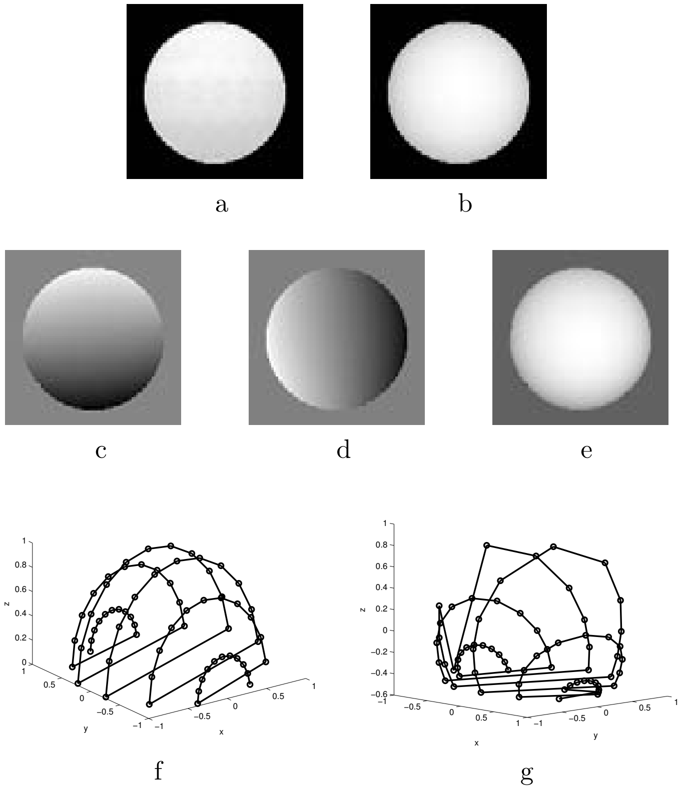


Figure 6.6: sphere images obtained from data generated with POV-Ray, shadows are considered as missing data, given a matrix with 57% of missing data; (top) recovered reflectance (a) and ambient illumination (b); (middle) x,y and z components of the recovered surface normals; (bottom) ground truth (f) and recovered (g) light position.

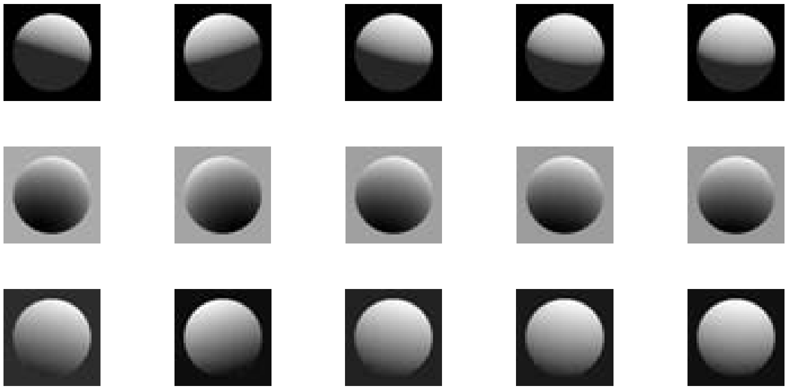


Figure 6.7: (top) a set of the original images generated with the POV-Ray, shadows are considered as missing data, given a matrix with 57% of missing data; (middle) images recovered by projecting the original ones onto a four-dimensional subspace; (bottom) images of the middle, adding the ambient term.

If the specular pixels (those for which the image intensity is equal to 255) are considered as missing data, the *measurement matrix I* has a percentage of missing data of about 28%. Fig. 6.10 shows the obtained results in this case. It can be seen that the ambient, reflectance and normals are considerably less specular than the recovered ones when considering 100% of data.

The difference between the full and the missing data case can be appreciated in Fig. 6.11. The recovered images are not as specular as the obtained in the full data case (Fig. 6.9).

Just as an illustration, Fig. 6.12 shows five images obtained by considering random positions of the light source. The normal surface matrix (S) obtained in the previous case (28% of missing data) is multiplied by each of the new light positions. The ambient term is added, in order to give more realistic images. This can be very useful, for instance in object recognition, industrial inspection, etc. Any desired light position can be taken, providing thus a wide range of different images.

Sculptured Bust Images

The images of this second real data set also contains saturated pixels (see examples in the first row in Fig. 6.14). The images have a size of 404×260 pixels and only 49 images are considered, given a measurement matrix of 65,246×49 elements. If the background pixels were also considered, the matrix would have 105,040 rows.

The obtained results with this data set are shown in Fig. 6.13. Some initial and recovered images are shown in Fig. 6.14. The original specular regions are a little specular in the recovered images. Actually, the saturated points influence more the results than the pixels in shadow, which are also present in these images.

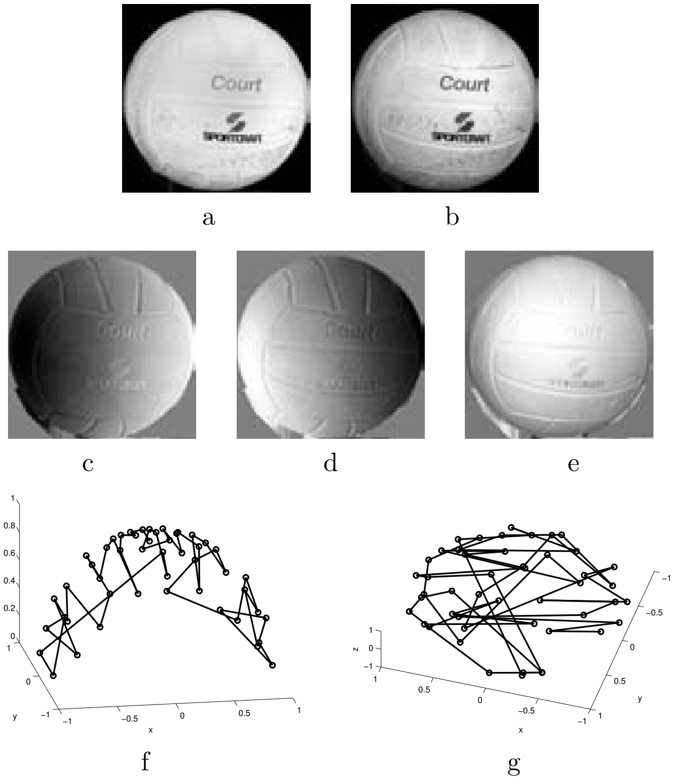


Figure 6.8: ball images; (top) recovered reflectance and ambient illumination; (middle) x,y and z components of the recovered surface normals; (bottom) ground truth and recovered light position.

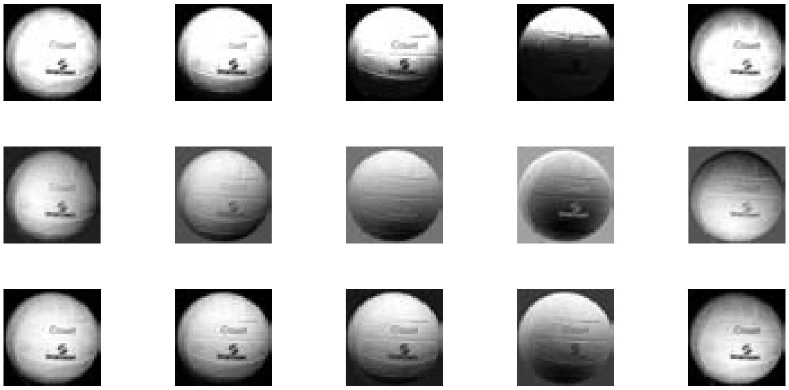


Figure 6.9: (top) a set of the original images of the ball; (middle) images recovered by projecting the original ones onto a four-dimensional subspace; (bottom) images of the middle, adding the ambient component.

If the saturated points are considered as missing data, the *measurement matrix* has a percentage of missing data of about 16%. Fig. 6.15 shows the obtained results in this case. The ambient (Fig. 6.15 (a)) and reflectance (Fig. 6.15 (b)) are less saturated than in the full data case.

Analogously, the recovered images are considerably less saturated than in the full data case, as it can be seen in the third row in Fig. 6.16.

6.2.6 Conclusions

This Section proposes the Alternation technique together with the corresponding adaptation to tackle the photometric stereo problem. The goal is to obtain the surface normals and reflectance of an object from a given set of images obtained under varying illumination. On the one hand, it is proposed to consider the entries of the matrix that correspond to saturated points or pixels in shadow in the image as missing data. On the other hand, a rank-4 formulation that allows to recover the ambient illumination is introduced. Experimental results with synthetic and real data show the viability of the proposed adapted Alternation approach. Furthermore, results are improved when shadows and specularities are considered as missing data.

The fact of working with matrices with large number of rows, makes it difficult to apply the iterative multiresolution scheme presented in Chapter 4. It would allow to deal with higher percentages of missing data.

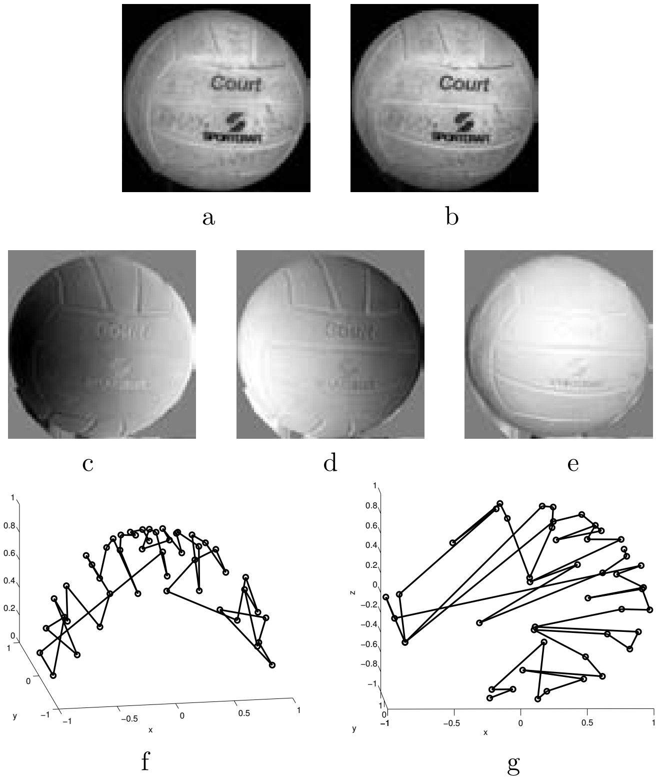


Figure 6.10: ball images, specularities are considered as missing data giving a percentage of missing of about 28%; (top) recovered reflectance and ambient illumination; (middle) x,y and z components of the recovered surface normals; (bottom) ground truth and recovered light position.

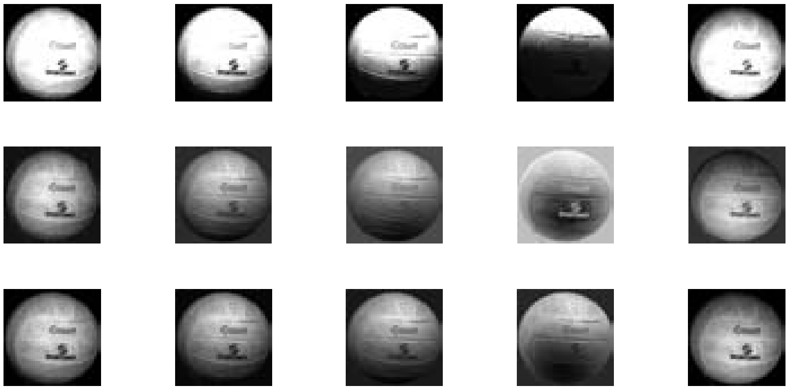


Figure 6.11: (top) a set of the original images of the ball, specularities are considered as missing data, giving a percentage of missing of about 28%; (middle) images recovered by projecting the original ones onto a four-dimensional subspace; (bottom) images of the middle, adding the ambient component.

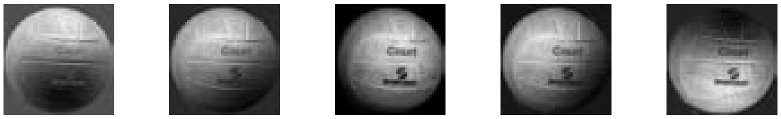


Figure 6.12: images synthesized by considering random light source positions.

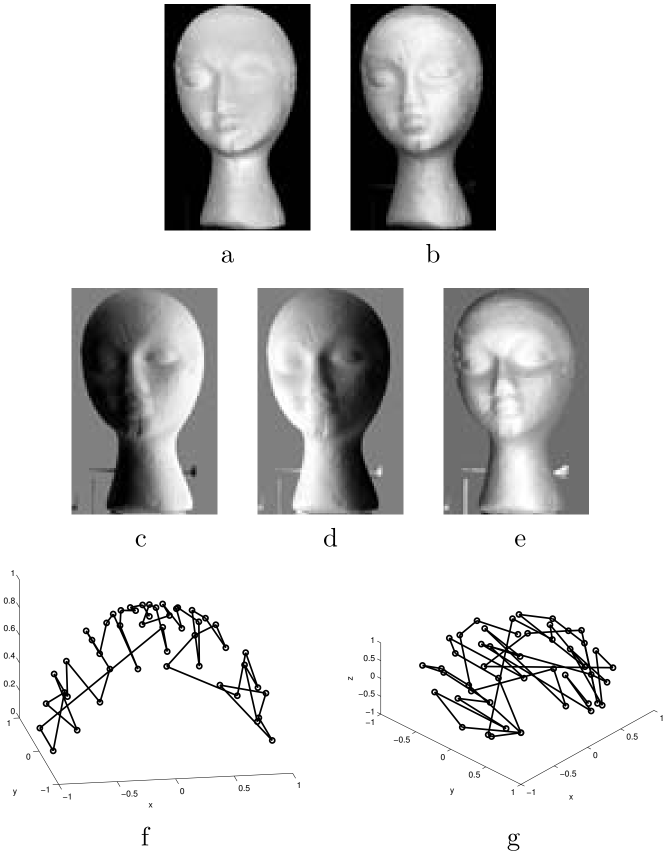


Figure 6.13: sculptured bust images; (top) recovered reflectance and ambient illumination; (middle) x,y and z components of the recovered surface normals; (bottom) ground truth and recovered light position.

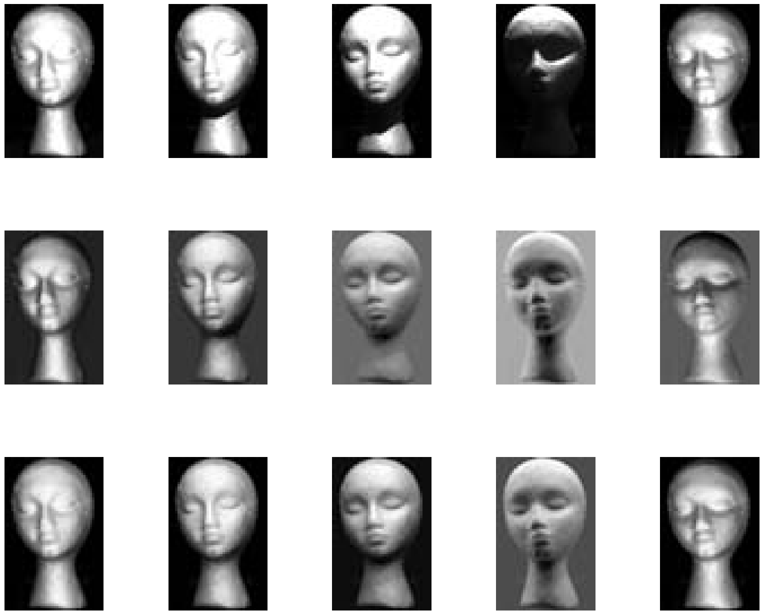


Figure 6.14: (top) a set of the original images of the sculpture; (middle) images recovered by projecting the original ones onto a four-dimensional subspace; (bottom) images of the middle, adding the ambient component.

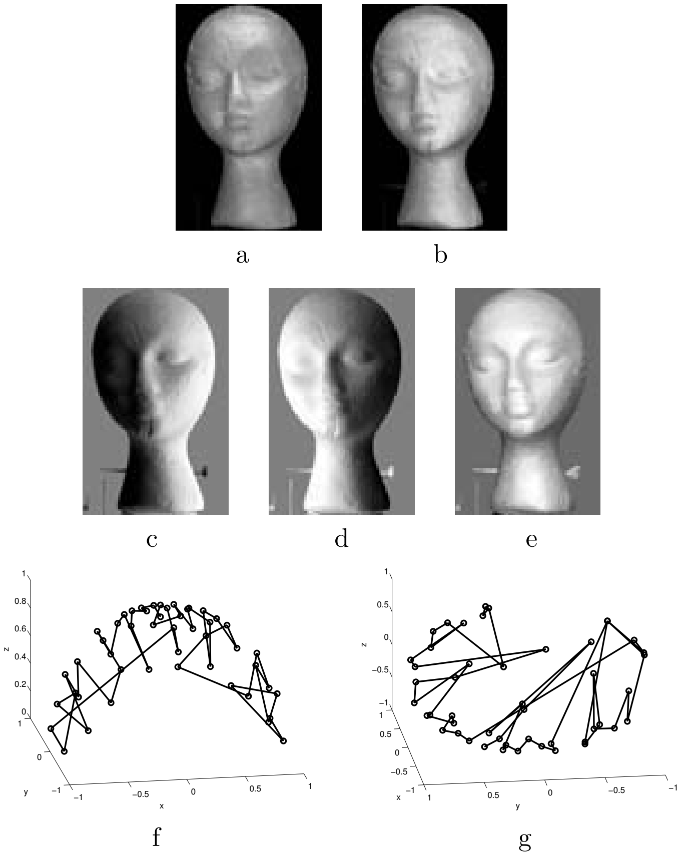


Figure 6.15: sculptured bust images, specularities are considered as missing, giving a percentage of missing data of about 16%; (top) recovered reflectance and ambient illumination; (middle) x,y and z components of the recovered surface normals; (bottom) ground truth and recovered light position.

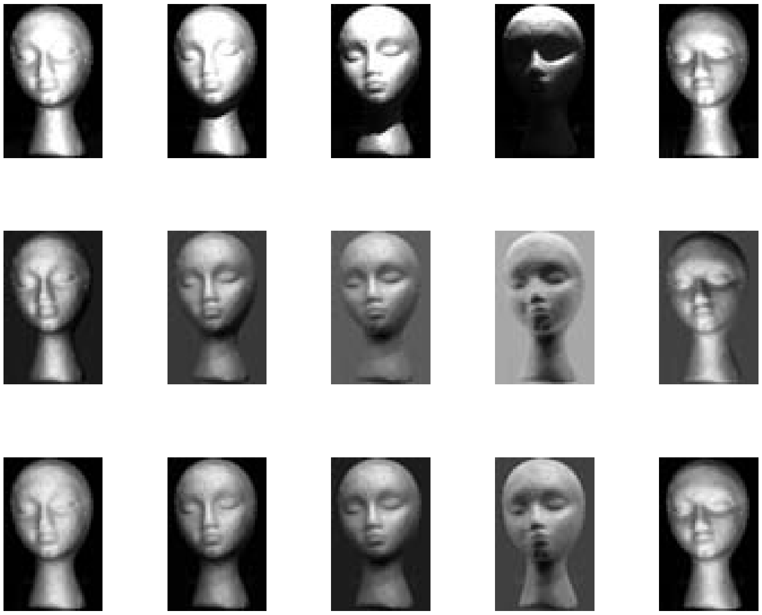


Figure 6.16: (top) a set of the original images of the sculptured bust, specularities are considered as missing data, giving a percentage of missing data of about 16%; (middle) images recovered by projecting the original ones onto a four-dimensional subspace; (bottom) images of the middle, adding the ambient component.

6.3 Missing Data Recovery in DNA Microarrays Using the Alternation Technique

As pointed out in [58], the mystery of life for a living organism resides in the function of thousands of genes. Traditional methods of analyzing gene expressions work studying a gene at a time, which is time consuming and costly. The *DNA*[2] *microarray* technology allows the whole genome to be monitored on a single chip so that a better picture of the interactions among thousands of genes can be observed simultaneously [19]. *DNA microarrays* are microscopic arrays of large sets of DNA immobilized on solid substrates. The data provided by *DNA microarrays* can provide fundamental understanding of life on the molecular level and may be useful in medical diagnosis, treatment and drug design [5]. However, as marked in [19], genome projects simply transfer digital information from DNA to computer files and that does not mean that everything about the molecular life can easily be known. The aim of the functional genomics science is to understand how the genome functions by controlling the expression of genes. Mathematical tools capable to deal with large amount of data, while reducing its complexity are required.

One of the methods used to obtain the gene expression is briefly described in [21]: arrays of thousands of discrete DNA sequences are printed on glass microscope slides using a robotic 'arrayer'. To compare the relative abundance of each of these gene sequences in two DNA or RNA[3] samples (for example, the total mRNA[4] isolated from two different cell populations), the two samples are first labelled using different fluorescent dyes (say, a red dye and a green dye). They are then mixed and hybridized with the arrayed DNA spots. After hybridization, fluorescence measurements are made with a microscope that illuminates each DNA spot and measures fluorescence for each dye separately; these measurements are used to determine the ratio, and in turn the relative abundance, of the sequence of each specific gene in the two mRNA or DNA samples. There are other microarray systems and methods that differ in many details, but they share the essential simplicity of this experimental design. Finally, the ratios are *log*-transformed (base 2) and stacked into large tables (matrices) of expressions levels of genes (rows) under different experimental conditions (columns). More information about the DNA microarray analysis can be found in [32].

Analysis of gene expression data aims at identify and cluster genes with similar expression patterns (e.g. [90], [5], [77]). The problem is that missing data are common in this kind of experiments and most of the developed clustering methods require a full matrix of gene expression values. The missing values may be due to insufficient resolution, image corruption, dust or scratches on the slide, or simply as a result of the robotic methods used to create them. The corrupted spots are manually or

[2] DNA: Deoxyribonucleic Acid. It is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms.

[3] RNA: Ribonucleic Acid. It is a nucleic acid consisting of many nucleotides that form a polymer. RNA plays several important roles in the processes of translating information from DNA into proteins.

[4] mRNA: Messenger Ribonucleic Acid. It is a molecule of RNA encoding a chemical "blueprint" for a protein product.

automatically flagged and corresponding signals are considered as missing, since they may influence wrongly the final results. Several algorithms for gene expression data analysis require a complete data matrix. Therefore, to impute the missing entries in the gene data matrix is a necessary pre-processing step. Furthermore, it is important to obtain correct imputed values. Otherwise, the analysis of the complete data could give incorrect or even contradictory conclusions.

Ignoring missing values is the simplest method to avoid working with missing data, and it is frequently applied. In most data sets, it supposes to reduce significantly the amount of data, and consequently, the quality of the analysis results. One possible solution to the missing data problem is to repeat the experiment, but this is a very expensive strategy. Other approaches consist in replacing the missing data by zeros or by an average expression over the row. Troyanskaya et al. [116] pointed out that these approaches do not consider the correlation structure of the data. Focusing on that problem, they present two different methods to estimate the missing values in the data matrix. The first one is based on the K Nearest Neighbour (KNN) and the second one is based on the singular value decomposition (SVD). They demonstrate the better performance of these two methods over the row average method and concretely, they conclude that the KNN-based method is more appropiated than the SVD-based one for recovering the missing values. In the KNN-based method, given a gene (denoted as *target* gene) with missing entries, K similar genes are selected. Then, each missing value in the target gene is estimated as the average of the corresponding entries in the selected K genes, weighted by a similarity measure (the Euclidean distance) between the target gene and each of the K selected genes. One drawback of this approach is that the number of neighbours (K) must be estimated. Oba et al. [88] propose an approach based on the Bayesian principal component analysis (BPCA). One of the main advantages of this approach is that no parameter is needed to be estimated. Friedland et al. [37] criticize the aforementioned iterative SVD-based approach, since its convergence is not guaranteed. They propose a method to estimate simultaneously all the missing entries in a gene expression matrix. That is, the estimation of one missing entry influences the estimation of the other missing ones. They define an optimization problem and present the *fixed rank approximation algorithm* (FRAA) to solve it. In a first step, the missing entries are filled in with zeros. Then, the solution is iteratively updated using SVD. In this case, the number of eigenvalues is needed to be estimated. Kim et al. [76] propose an imputation method based on the least squares formultation (LLSi). The missing values of a particular gene, which corresponds to a row in the data matrix, are estimated as a linear combination of values of other genes using standard least square regression. Two similarity measures are defined in order to select the most similar genes: the L_2-norm and the Pearson correlation coefficient. As in the KNN approach, the number of similar genes must be known. In a more recent approach, Friedland et al. [38] propose the *improved fixed rank approximation algorithm* (IFRAA), which is a combination of FRAA and a clustering algorithm. Firstly, FRAA is applied to fill in the missing values in the gene data matrix. Then, a clustering algorithm is applied in order to group similar genes. Finally, FRAA is applied to each of these groups of genes individually. In [101], Scheel et al. propose to study the influence of imputation on the detection of differentially expressed genes from microarray data (instead of computing the $nrms$, defined later) to measure the goodness of the imputed data. Furthermore, the authors propose a new method for imputation (LinImp), which consists in fitting a simple linear model for each channel (red and green) separately. Hu et al. [59] propose a method to improve the results obtained with LLSi and KKN. The key point is to incorporate information from multiple reference microarray datasets to improve missing value estimation.

This section propose to adapt the Alternation technique, introduced in Chapter 3, to impute missing data in gene expression data matrices. In particular, given a gene data matrix W, the product of the obtained factors A and B, are used as imputation: $W_{imputed} = AB$. Hence, initially known data are modified and, at the same time, missing data are recovered. This does not occur in the aforementioned approaches proposed for DNA microarray imputation, in which initially known data do not change. Although Alternation is not a novel technique, it has not been applied to fill in missing entries in DNA microarray, as far as we are concerned. Other factorization approaches have been used to study the expression levels of genes. For instace, in [77] the non-negative matrix factorization is applied to a full gene data matrix in order to obtain clusters of similar genes.

The Alternation aims at obtaining the best rank-r approximation of a given matrix, being r smaller than the dimensions of this matrix. However, in the current problem, the main drawback of applying the Alternation technique is that, in general, the rank of the gene data matrix is the number of its columns. In that case, the initially missing entries can be not properly recovered, as explained later. An Alternation-based strategy is proposed in order to obtain, on the one hand good recovered missing data and, on the other hand similar values to the original ones in the case of initially known data.

This Section is structured as follows. First, the proposed Alternation-based strategy, focused on the problem of imputing missing data in DNA microarray, is presented. Then, the data sets used in the experiments are introduced. In order to show the performance of the Alternation-based strategy, experimental results are provided. Finally, concluding remarks are summarized.

6.3.1 Alternation-based Strategy for Filling in Missing Data in DNA Microarrays

As detailed in Chapter 3, given an input matrix $W_{n \times p}$, the Alternation technique aims at finding the best r-rank approximation matrix to W, being $r < n, p$. In fact, the goal is to find $A_{n \times r}$ and $B_{r \times p}$ such that minimizes: $\|W - AB\|_F^2$. Working with missing data, the expression to minimize is:

$$\|W_{ij} - (W_{imputed})_{ij}\|_F^2 \tag{6.31}$$

where $W_{imputed} = AB$ and i and j correspond to known entries in W.

The algorithm converges rapidly if W is close to a rank-r matrix. Furthermore, it converges to a global minimum in the complete data case and in the case of a low

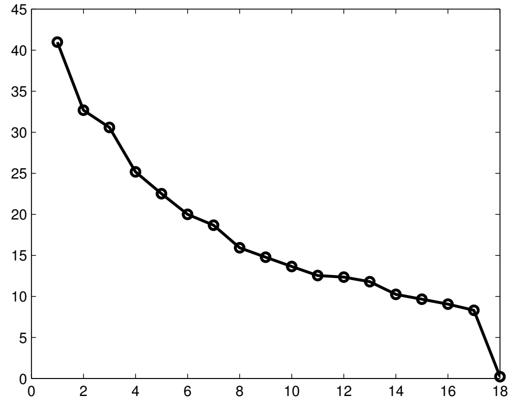


Figure 6.17: singular values of the initial matrix.

percentage of missing data [54]. With a large amount of missing entries in W, the algorithm may fail to converge, or converges very slowly.

An important fact that should be taken into account is that matrices containing gene expression data have, in general, the maximum rank, which is the number of columns of $W_{n\times p}$ ($p < n$, normally). In other problems, such as SFM, the rank of W is also the maximum due to the noise in the data. However, since many singular values are close to zero, a *real* rank smaller than p can be estimated. In the current problem, the singular values are not close to zero. An example of the singular values obtained from a full gene data matrix of size 4489×18 is plotted in Fig. 6.17. It can be seen that only the last singular value is negligible (its value is 0.21). This characteristic of gene data matrices makes difficult to obtain good results with the Alternation, as it is explained below.

The Alternation technique gives a good rank-r approximation to W, while the number of known entries at each row/column is at least r. Otherwise, the initially missing entries are wrongly estimated. This is due to the fact that, at each step of the Alternation algorithm, the number of unknown entries (each row of A or column of B) is r (see Chapter 3 for details). The number of equations to compute these unknowns (referred to as n_e from now on) is the number of known elements in the corresponding row of $W_{n\times p}$ if A is computed (or corresponding column when B is computed). If $r = p$ (as in the studied gene data matrices), with only a few missing entries, the number of equations is smaller than the number of unknowns ($n_e < r$) and the algorithm may not converge to a global minimum. Therefore, the obtained p-rank approximation to W might not be correct. Concretely, the initially missing entries may be quite different from the correct ones.

One possible solution would be to approximate W by a r'-rank matrix, with $r' < r$. Thus, if $n_e \geq r'$, the unknown entries would be well estimated. Unfortunately, if $r' < r$, the initial known entries would not be as well approximated as taking full rank. Hence, it is difficult to minimize the error corresponding to the estimation of missing entries (r should be smaller than p) and, at the same time, the error corresponding to initially known data (r should be equal to p).

In order to tackle this problem, an Alternation-based strategy is proposed. It consists in applying the Alternation to the matrix with missing data, taking $r = 1$. With this very low rank value, only one known entry per row/column of W is required in order to obtain properly recovered missing data. However, the initially known entries are not correctly recovered, since the rank is far from the real value (p). Next step consists in applying the Alternation to the initial matrix again, but taking $r = 2$ and replacing the missing data with the corresponding recovered data in the previous step. Hence, there are missing data only in the first step. The initially missing entries are better estimated as r increases, since the rank gets closer to the correct value. This process is repeated until the rank reaches the number of columns (p), in which case the error computing the initially known entries is close to zero. The proposed strategy is summarized below.

Alternation-based strategy: given a matrix with missing data $W_{n\times p}$, two weight matrices that contains only ones and zeros are defined: the first matrix, denoted as $\mathbf{M}$, represents the missing entries positions in W. Concretely, the ones and zeros in $\mathbf{M}$ correspond to the missing and known entries in W, respectively. The second matrix, denoted as $\mathbf{K}$, represents the known entries positions in W. In this case, the ones and zeros in $\mathbf{K}$ correspond to the known and missing entries in W, respectively.

Set $r = 1$, $W_0 = W$ and repeat the following steps until $r = p$:

1. Apply the Alternation technique to W_0, taking the current rank r. A full matrix is obtained with the product of the recovered factors: $W_{imputed} = AB$

2. If $r < p$, define:

$$W_r = \mathbf{M} \odot W_{imputed} + \mathbf{K} \odot W \tag{6.32}$$

   where $\odot$ is the Hadamard product[5]. That is, with the first summand ($\mathbf{M} \odot W_{imputed}$), only the entries of $W_{imputed}$ corresponding to missing entries in W are selected. The rest of the entries are equal to zero. In the second summand ($\mathbf{K} \odot W$), only the entries corresponding to the known entries positions in W are selected. Thus, entries corresponding to missing data in W_0 are replaced by the imputed values in the corresponding positions and the initially known entries do not vary. Set $W_0 = W_r$ and $r = r + 1$.

   Otherwise, $W_r = W_{imputed}$ is the complete matrix obtained with the Alternation-based strategy.

Solution: W_r, for $r = p$. On the one hand, the known entries in W_r are very close to the initially known values in W, since $r = p$. On the other hand, the best values

[5] $R = P \odot Q \Longleftrightarrow r_{ij} = p_{ij}q_{ij}$

for the missing entries are obtained. The key point of this strategy is that there are missing data only in the first step of the algorithm. Therefore, there is no problem about the number of known entries in each row/column.

6.3.2 Data Sets

As in the SVD-based [116] and BPCA-based [88] approaches, in the Alternation technique it is assumed the existence of a covariance structure among all the genes or samples in the data matrix. Hence, these three methods are more suitable for data sets with strong global correlation, such as time-series data sets.

Therefore, time-series data sets are used in our experiments. Concretely, three different data sets are considered. The first data set is the one presented in [105], where Spellman et al. aim at creating a catalog of yeast genes. It is publicly available at: http://cellcycle-www.stanford.edu/. The authors present results of a series of experiments, focused on identifying all protein-encoding transcripts in the genome of the yeast *Saccharomyces cerevisiae*. This data set consists of *log*-transformed (base 2) ratios of the intensities obtained in the hybridization process and it is also used in [116], [88], [76], [37] and [59]. Two different groups, which give two different data matrices, are obtained from this data set: W_{alpha} and W_{elu}. The alpha part consists of 6178 genes (18 samples). However, if only genes without missing data are considered, a total of 4489 genes are selected. The elutriation part contains 6178 genes and 14 samples, although only 5766 genes do not present missing data in the experiments.

The second data set is an exploration of temporal gene expression during the metabolic shift from fermentation to respiration in *Saccharomyces cerevisiae* [31]. It is available at: http://derisilab.ucsf.edu/data/hotspots/. It is also used in [116] and [59]. In this case, data are the ratios of the intensities and the obtained matrix contains 6183 gens and 7 experiments.

Finally, the third data set is the one studied in [40]. Again, the genome of *Saccharomyces cerevisiae* is studied. It consists of the obtained intensities in the hybridization process, instead of the *log*-transformed ratios and it is publicly available at: http://cmgm.stanford.edu/pbrown/explore/index.html/. Three different data matrices can be generated: the matrix that contains the intensities, the matrix of the ratios of these intensities and the matrix of the *log*-transformed (base 2) ratios. The goal is to study if the results depend on way the data are represented (intensities, ratios or *log*-transformed ratios). These matrices consist of 6153 genes and 7 experiments. Only the first 5000 genes are considered in the experiments, for simplicity.

6.3.3 Experimental Results

This Section contains the experimental results. Although several approaches have been proposed in he literature, only the most recent and remarkable ones are used to compare with the proposed strategy. Concretely, the performance of the proposed Alternation-based strategy (referred to as ALT hereinafter) is compared with the KNN-based approach presented in [116], the BPCA [88], the LLSi [76] and the FRAA [37]. KNN is the classic approach and it is studied in most of the papers. BPCA and LLSimpute approaches are also studied in the literature and give, in general, good results. Furthermore, publicly available implementations of both algorithms are provided at: http://hawaii.aist-nara.ac.jp/∼shige-o/tools/ , in the case of BPCA [88] and at http://www.cs.umn.edu/∼hskim/tools.html , in the case of LLSi [76]. Finally, FRAA is one of the most recent proposals and the Matlab code of the algorithm is provided in [37].

Other approaches proposed in the literature are not tested: a) SVDimputation, presented in [116], is not studied, since posterior approaches outperform it; b) to implement the approach presented in [59] it is considered no necessary, since the KNN and the LLSi performs better than [59]; c) finally, the approach proposed in [38] is not tested since the clustering of genes is out of the scope of this work.

Although in some cases the original data matrix $W_{original}$ has missing data, a complete matrix W_{full} is considered in order to study the goodness of the recovered data. Missing entries are generated artificially as detailed in short. In most presented approaches the original known data do not vary. That is, missing data are recovered, but known data do not change. The most common measure of goodness used in DNA is the *normalized root mean squared error* (*nrms*), which is defined as follows:

$$nrms = rms_{missing}/\sqrt{variance((W_{full})_{ij})} \tag{6.33}$$

where i and j correspond to the index pairs where W_{ij} has missing data and the *root mean squared error* ($rms_{missing}$) is defined as:

$$rms_{missing} = \frac{\|W_{full} - W_{imputed}\|_F}{\sqrt{n}} = \sqrt{\frac{\sum_{i,j} |(W_{full})_{ij} - (W_{imputed})_{ij}|^2}{n}} \tag{6.34}$$

Notice that W_{full} represents the complete matrix and $W_{imputed}$ represents the filled in matrix with an imputation technique. It should be remarked that this measure is different to the usually defined *rms* (e.g., equation (4.9) , in Chapter 4), in which only the initially known entries are considered, instead of the initially missing ones.

Different percentages of missing data are considered. Concretely, the studied missing percentages are 1%, 5%, 10%, 15% and 20%. In most approaches proposed for missing data imputation in DNA microarrays, missing data are generated following a random distribution (e.g., [116], [76], [37], [59]). Although most of the literature does not take it into account, the missing data can be not at random, as pointed out in [88] and [101]. As an example, some signals could not be distinguished from the background or have a too irregular form because the signal itself is too low. In these cases the missing data structure depends on the signal intensity, which can follow a non-random distribution. Actually, the most common situation is a mixture of random and non-random distributions. Different proposals have been presented in the literature, in order to study more realistic situations.

Oba et al. [88] propose a *histrogram-based* way of obtaining the missing data. They study the histogram of column-wise numbers of missing entries in the original matrix. Then, data are removed from the complete matrix (obtained by discarding the genes with missing data) so that the histogram of the artificial missing entries is similar to the histogram of the original missing entries. This is not easy to reproduce when the reduced complete matrix has very small dimensions compared to the original matrix.

Scheel et al. [101] consider as missing entries the values corresponding to low intensities. Notice that they work with intensities instead of with *log*-transformed ratios. They propose to sort the signals and consider a percentage of the lowest intensities. Then, a smaller percentage of this first set is removed randomly.

Friedland et al. [38] propose to create the missing data by taking the original missing data positions and consider as missing entries the corresponding positions in the full data matrix. Unfortunately, as far as we are concerned, this strategy is only possible if the complete matrix is very large. However, even in that case, the exact positions of the missing data do not correspond to the same positions in the complete matrix, which has smaller dimensions. Actually, the obtained matrix has a percentage of missing data higher than the original one.

In the current experiments, two different missing data structure are studied. On the one hand, random missing data are considered. On the other hand, a combination of random missing and *line-missing*, which is common in real situations, is proposed. This concrete structure of missing data is referred to as non-random missing from now on. In real data, it is common to find a gene that has missing entries nearly in all the experiments. For instance, when a gene presents a missing entry due to low intensities, the same gene presents several missing entries, given rise to a nearly empty row in the data matrix. This is what we call *line-missing*. In Fig. 6.18 (left), a real example of a missing data matrix is shown. Missing data are represented as white entries. Notice that there are genes (rows in the matrix) that have many missing entries. Fig. 6.18 (middle) and Fig. 6.18 (right) show two examples of gene data matrices where a 10% of missing has been obtained following the random and non-random procedures mentioned above.

Random Missing

The results corresponding to the first data set are shown in Fig. 6.19 for the matrix W_{alpha} and Fig. 6.20 for the matrix W_{elu}. It can be seen that, in general, the smallest error is obtained with the BPCA approach. The LLSi gives similar *nrms* values to the ones given by BPCA, even smaller when the percentage of missing data is below 5%. However, when the LLSi is applied, a problem should be faced out:

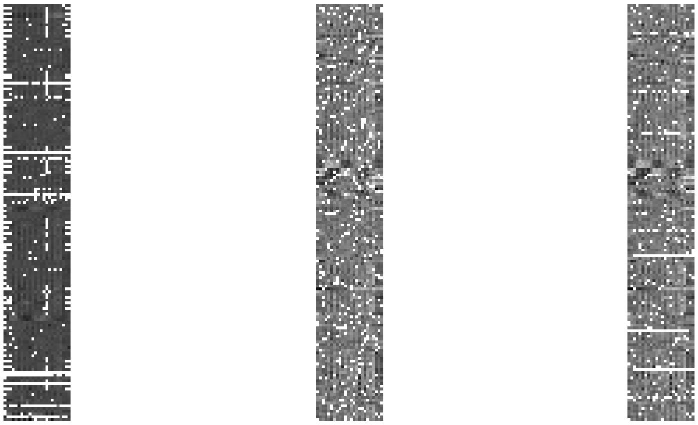


Figure 6.18: (left) real missing data in a gene data matrix; (middle) gene data matrix with 10% of random missing data; (right) gene data matrix with 10% of non-random missing data.

the number of similar genes used (K) is needed to be estimated. The same problem is found in the KNN approach: the number of neighbours (K) used to estimate the missing entries is needed to be known. Finally, with the FRAA approach, a different parameter must be estimated: the number of significant eigenvalues (L) in the gene data matrix. The *nrms* values obtained with ALT are similar to the KNN, a little smaller with percentages of missing data below 15%. The advantage is that any parameter has to be estimated with ALT.

It should be remarked that several possible K (and analogous for L) values are considered in the experiments and the one for which the *nrms* is smaller is plotted. However, in this parameter selection process, the unknown data values are used. Notice that this can be done, because the missing is artificially generated. In real situations, these missing values are really unknown and the parameters should be fixed beforehand. As illustration, the *nrms* values obtained considering different parameter values and with a percentage of missing data of 5%, are plotted in Fig. 6.19 (right) and Fig. 6.20 (right). The results of the approaches that do not depend on any parameter (BPCA and ALT) are shown on the y-axis (a similar plot is presented in [76]). Notice that although LLSi outperforms the proposed technique (see Fig. 6.19 (left) and Fig. 6.20 (left)) it is quite sensitive to the user defined parameter (K) (see Fig. 6.19 (right) and Fig. 6.20 (right)). In the KNN case, the results are not as sensitive to the K value as in the LLSi, as it can be seen in Fig. 6.19 (right) and Fig. 6.20 (right). The highest *nrms* is obtained with the FRAA approach, in both examples. As it can be appreciated in Fig. 6.19 (right) and Fig. 6.20 (right), the smallest *nrms* is obtained when $L = 5$. The *nrms* becomes higher as the L value increases.

Fig. 6.21 shows the *nrms* obtained with the second data set. Recall that in this case the data matrix consists of ratios of intensities obtained in the hybridization process. The LLSi gives in general the smallest *nrms*, although similar results are

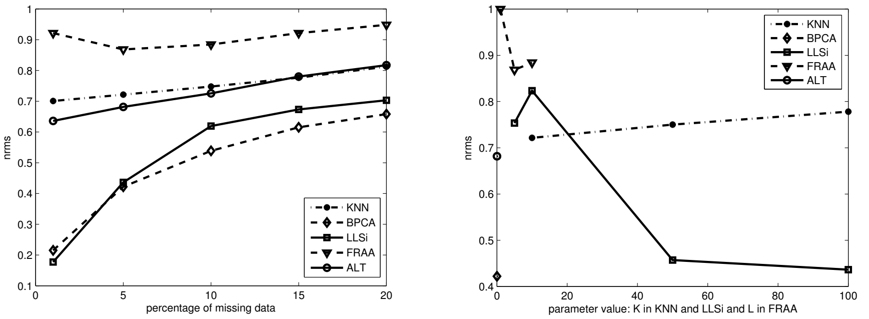


Figure 6.19: W_{alpha}; (left) *nrms* for different percentages of random missing data; (right) *nrms* for different tested parameters: K in the case of KNN and LLSi, and L in the case of FRAA. The percentage of random missing data is 5%.

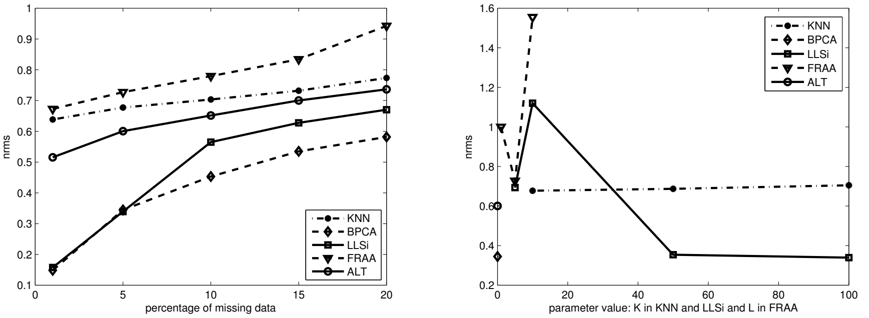


Figure 6.20: W_{elu}; (left) *nrms* for different percentages of random missing data; (right) *nrms* for different tested parameters: K in the case of KNN and LLSi, and L in the case of FRAA. The percentage of random missing data is 5%.

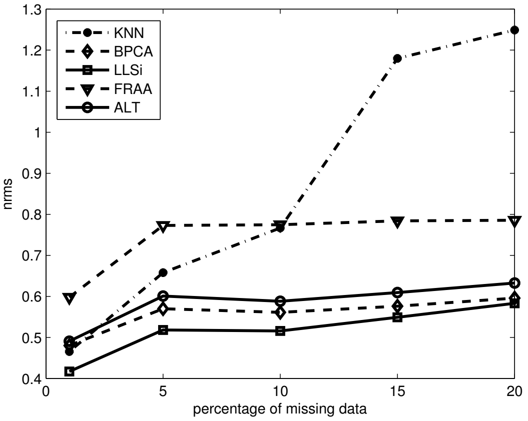


Figure 6.21: second data set; *nrms* for different percentages of random missing data.

obtained with BPCA and ALT. This is due to the fact that, as pointed out in [76], this approach is the most appropriated working with only a few experiments (number of columns in the gene data matrix). In this case, there are only 7 different experiments. The *nrms* obtained with the KNN grows as the percentage of missing data increases. In particular, the highest *nrms* is obtained with the FRAA, while the percentage of missing data is below 15%, and with the KNN with more missing data.

The *nrms* values obtained with the third data set are plotted in Fig. 6.22 and Fig. 6.23. As in the previous example, the LLSi gives in general the smallest *nrms*. Again, the gene data matrix has only 7 experiments. Fig. 6.22 shows the obtained *nrms* in the case of the data matrix containing the intensities. The LLSi, BPCA and ALT give similar *nrms* values, while the FRAA gives higher *nrms* values. As in the previous example, the *nrms* values in the KNN case grows remarkably as the percentage of missing data increases.

The results obtained considering a matrix of ratios are plotted in Fig. 6.23 (left). On the other hand, Fig. 6.23 (right) shows the obtained results when the *log*-transformed ratios are considered. The *nrms* values obtained with the ALT are a little higher than those ones obtained with the BPCA and LLSi. In both cases, the BPCA gives smaller *nrms* than the LLSi when the percentage of missing data is higher than 10%. Again, the *nrms* obtained with the KNN grows as the percentage of missing data increases. The FRAA gives, in general, high *nrms* values, but it is not as sensitive as the KNN to the percentage of missing data.

It should be noticed that the *nrms* values increase, in general with any method, with the matrix of ratios and also with the *log*-transformed ratios matrix. This is due to the fact that the matrix containing the intensities has a high variance, giving a small *nrms*. On the contrary, the variance of the matrix of ratios and the one of the *log*-transformed ratios is very small (smaller than 1) and the resulting *nrms* is

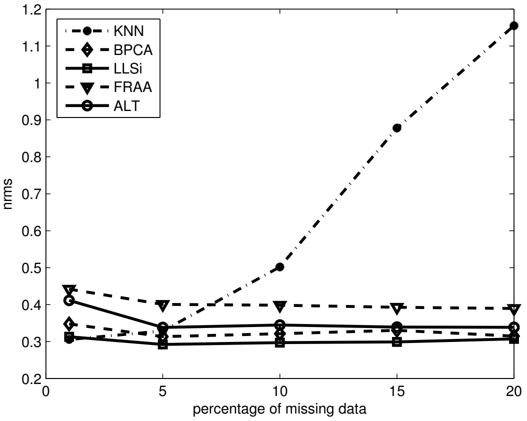


Figure 6.22: third data set, matrix of intensities; *nrms* for different percentages of random missing data.

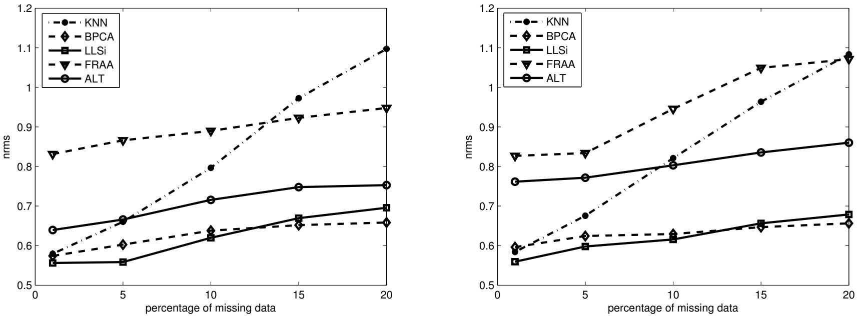


Figure 6.23: third data set, *nrms* for different percentages of random missing data; (left) matrix of ratios; (right) matrix of *log*-transformed ratios.

higher. Notice that, in the case of ALT, this increase is more marked.

Non-random Missing

The results obtained with the first data set are plotted in Fig. 6.24. Notice that, as in the random missing case, several possible K (and analogous for L) values are considered in the experiments and the one for which the *nrms* is smaller is plotted. The BPCA gives the smallest *nrms*, for both W_{alpha} (Fig. 6.24 (left)) and W_{elu} (Fig. 6.24 (right)). Although the *nrms* is higher than in the random missing case for all the methods, the increase is more significant in ALT and, even more in the KNN approach. In fact, in this last case, the KNN gives higher *nrms* than the FRAA approach. Hence, these two approaches are more sensitive to the distribution of missing data than the other approaches.

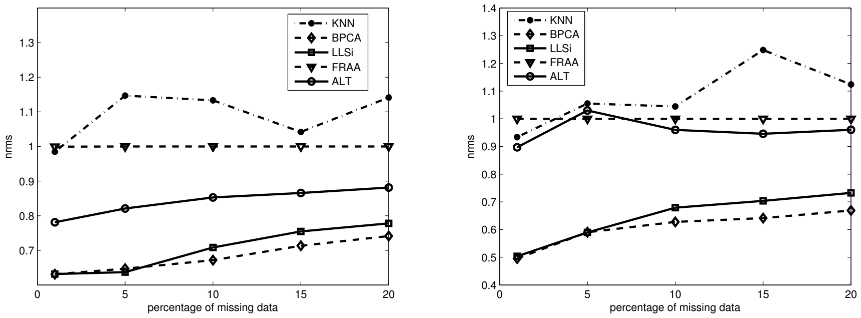


Figure 6.24: (left) W_{alpha}; *nrms* for different percentages of non-random missing data; (right) W_{elu}; *nrms* for different percentages of non-random missing data.

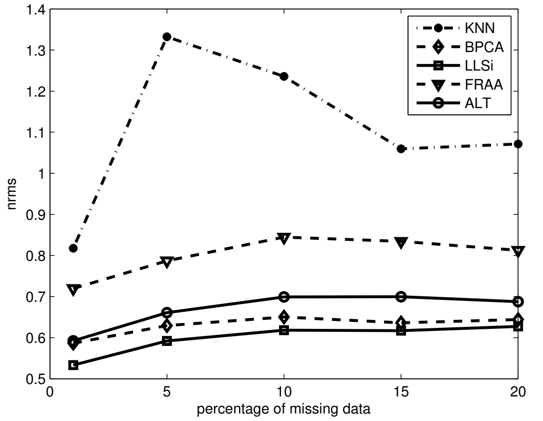


Figure 6.25: second data set; *nrms* for different percentages of non-random missing data.

Fig. 6.25 shows the *nrms* obtained with the second data set. The obtained *nrms* values are similar to the ones obtained with the random missing data case. As before, the smallest error is obtained with the LLSi and similar results are obtained with BPCA and ALT. The *nrms* values obtained with the KNN are very high (even selecting the K value that gives the smallest *nrms*).

The obtained *nrms* in the third data set are plotted in Fig. 6.26 and Fig. 6.27. In the particular case of the matrix of intensities (see Fig. 6.26), the *nrms* values are similar to the ones obtained with a random missing data distribution. The smallest *nrms* is obtained with the LLSi. However, similar results are obtained with the BPCA and the ALT. Again, the highest error is obtained with the KNN.

Fig. 6.27 (left) shows the results obtained in the case of the matrix of ratios. The *nrms* values are again similar to the ones obtained in the random missing data case. When the matrix of *log*-transformed ratios is considered (Fig. 6.27 (right)), the

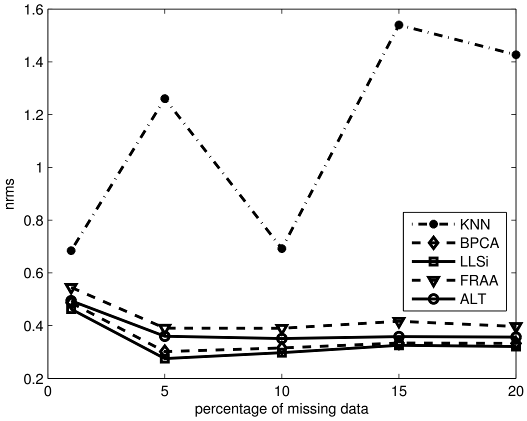


Figure 6.26: third data set, matrix of intensities; *nrms* for different percentages of non-random missing data.

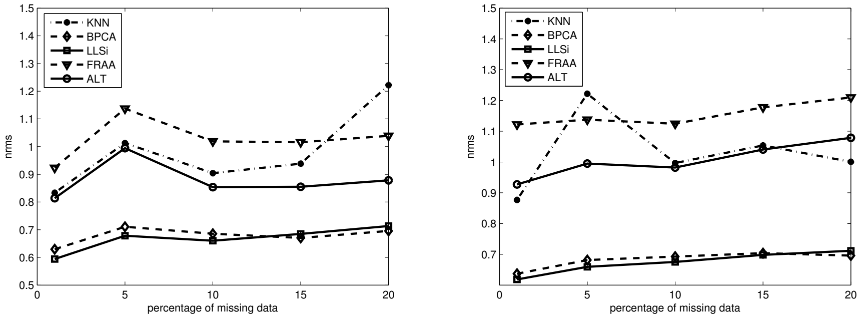


Figure 6.27: third data set, *nrms* for different percentages of non-random missing data; (left) matrix of ratios; (right) matrix of *log*-transformed ratios.

obtained *nrms* values are in general higher than the ones obtained in the random missing data case.

As in the random missing case, the *nrms* values obtained with the matrices of ratios and *log*-transformed ratios are higher than the obtained with the intensity matrix. Again, the difference among the *nrms* values corresponding to different approaches is more significant in the case of the ALT.

6.3.4 Conclusions

The use of an Alternation-based strategy is proposed for imputing missing entries in DNA microarray data. Although the Alternation is not a novel approach, it has not been used to tackle this problem, as far as we are concerned. Working with missing data, it is difficult to obtain results that minimize the error corresponding to the initially known entries and, at the same time, the error corresponding to the initially missing ones, since the gene data matrices are in general of full rank.

Comparatives of the performance of the proposed Alternation-based strategy and other methods are given. It is shown that, in most cases, other existing methods, such as BPCA and LLSi, outperforms the Alternation-based strategy. Concretely, LLSi performs better than the BPCA when the number of experiments is very small. However, with the LLSi, it is necessary to estimate the number of similar genes used to complete the missing entries, K. Furthermore, LLSi is quite sensitive to K selection. If the number of experiments is not small, the BPCA-based approach performs better than the other approaches. Furthermore, it is not necessary to estimate any parameter. On the other hand, the FRAA gives the highest errors in most of the experiments. In addition, a parameter (number of significant eigenvalues) must be known. The KNN gives errors higher than the BPCA, LLSi, and, working with only a few experiments, also higher than the FRAA. Besides, the number of neighbours K needs to be estimated. One advantage of this approach is that it is not as sensitive to the number of neighbors as the LLSi. Furthermore, the error value does not depends so much on the percentage of missing data.

The Alternation-based strategy, gives in general better results than the FRAA and the KNN, but not as good results as the BPCA and the LLSi. One advantage of the Alternation-based strategy, is that it is not necessary to estimate any parameter (not even the rank of the data matrix). Another advantage of the Alternation-based approach with respect to the other proposed techniques is that it gives two factors that contain the eigen-genes and the eigen-experiments. Other approaches aim to obtain this decomposition by applying the SVD and use it to cluster similar genes or experiments. This problem has not been tackled in this work.

Unfortunately, in this current application, each experiment is independent from the others and the matrix has a full rank. One possible strategy to obtain a non-full rank matrix could be to enlarge the matrix with linearly dependent columns. For instance, repeating the experiments would give dependent columns and the missing entries would probably be in other rows. The problem is that this solution would be a very expensive task. Furthermore, it would have no sense, since the results obtained with other approaches, such as BPCA, are acceptable. Therefore, as a conclusion, we think that the Alternation is a method more appropriated for problems with smaller dimension.

The iterative multiresolution scheme proposed in Chapter 4 is not used. The main reason is that it is an approach useful when there is a high percentage of missing data. In gene data matrices, the percentage of missing data is in general of 20% at most.

6.4 The Alternation Technique Applied to Recommender Systems

It is well known that the amount of information available on the Web increases constantly. Due to that fact, sometimes it becomes difficult to search for interesting information, while discarding useless content. Therefore, there is a clear demand for methods that give information with respect to users's preferences. At the same time, the number of queries is also increasing. For instance, information about a wide variety of preferences is available. This information can be used to provide recommendations to new users. This is the goal of recommender systems: they aim at helping users to find items they would probably like or dislike using only a few given preferences. That is, users give rates only to some items and the system is capable of predicting their punctuation to the rest of items (this is called *prediction task*). Furthermore, it can also recommend them the products they would probably like most (*recommendation task*). These two powerful tools are widely used on *e-commerce* Web sites. Since their introduction in 1990s, the recommender systems have been used to filter information on the Web and to provide recommendations about books, CDs, movies, news, electronics, financial services, travel, etc. One of the most popular recommender system is the one used at: http://www.amazon.com. The customer rates some books and the system suggests him other books using information from other customers. Another recommender system can be found at: http://www.cdnow.com. In this case, the recommended products are CDs instead of books. One different recommender system is used at: http://www.everyonesacritic.net, where users give their opinion about movies and the system makes recommendations from people who share similar tastes. One last example (www.gnomoradio.org) consists of a music recommender system. The user can rate the music and the program tries to build a listening profile based on the user's ratings. In addition, it recommends music that other users with similar profiles like. In fact, in most cases, the main goal of the recommender systems is to discover which products will like a customer at most, in order to increase the sales and therefore, the benefits. Of course this helps also the customer, who will find, in theory, something useful to him.

In recommender systems, data are stored into a large table of users (also denoted as customers) and items (or products). Actually, the information is stored into a matrix of data, whose rows and columns correspond to each user and item respectively, and whose entries correspond to the rates that customers give to items. In real problems, there is a wide set of both customers and items. Hence it is necessary to deal with large matrices. Furthermore, since each user only rates a subset of the large set of items, most entries of the data matrix are empty and the matrix tends to be very sparse.

One technique widely used in recommender systems is the *collaborative filtering* (e.g., [94], [135], [42]), which is usually based on finding neighbourhoods of *similar* customers that are obtained by computing the correlation between their opinions. The similarity function is different in each approach. Although this technique is useful in many domains, it has a high computational cost and has limited prediction when dealing with very sparse data, as pointed out in [99] and [15]. In fact, Billsus et al. [11] identify two important limitations in the collaborative filtering techniques. The first one is that the correlation between two users rates can only be computed on items that both users have rated. Since, in general, there are thousands of items to rate, the number of overlapped items is small in most cases and the similarity measure is based on the correlation of only a few items. The second problem is that with this similarity measure, two users can only be similar if there is overlap among the rated items. As mentioned above, when the number of items to rate is large, it is difficult to obtain overlap among the rates.

Billsus et al. [11] present collaborative filtering in a machine learning framework to tackle the aforementioned limitations. Their proposed approach is based on the Singular Value Decomposition (SVD) [43]. Other recommender systems used the SVD (e.g., [92], [10], [99]) to reduce the data representation and give predictions using linear regression. The main advantage of the SVD is that, not only the information of correlated customers are used, but also the obtained from users whose ratings are not correlated, or who have not even rated anything in common. The SVD allows to project user ratings and rated items into a lower dimensional space. Thus, some users become predictors for other users's preferences even without any overlap of rated items. Unfortunately, computing the SVD of a large matrix requires a high computational cost. Furthermore, all the data must be known. Therefore, in order to be able to apply SVD, the missing data must be filled in. Some approaches add zeros in the missing entries, while others fill them with the corresponding row or column average (e.g., [99]). Then, these previously filled in missing entries are actualized with the SVD.

A different approach, based also in the SVD, is proposed by Brand in [15]. Actually, it is an imputation method developed in [14] to predict the position of occluded features in computer vision problems. In [15], it is used in data mining tasks. Concretely, Brand presents a method for adding data to a *thin* SVD[6] data model, which is significantly quicker and economical than full SVD. Instead of computing the SVD of a large matrix, he develops an exact rank-1 update which provides a linear-time construction of the whole SVD. The approach begins by sorting out the rows and columns of the data matrix so that a high density of data is accumulated in one corner. Then this initial full submatrix grows out of this corner by sequential updating with partial rows and columns. An imputation update that maximizes the probability of correct generalization is used to fill in the missing entries. Furthermore, they give a very interesting interpretation of the factors obtained with the SVD. In particular, these obtained factors are the r dimensional representations of the customers and the items, respectively. These two representations can be very useful for grouping customers who like/dislike the same items, or grouping factors by the sorts of customers who like them, etc. Many interesting applications for marketing could be derived from this subspaces' interpretation. One disadvantage of this technique is that, as pointed out in [54], the result depends on how the data are sorted.

[6] Given a matrix $W_{m\times n}$, being $n \ll m$, the *thin* SVD consists in computing only the n column vectors of U corresponding to the row vectors of V^t. That is: $W_{m\times n} = U_{m\times n}\Sigma_{n\times n}V^t_{n\times n}$. The remaining columns of U are not computed

In addition to the aforementioned limitations, one problem of the collaborative filtering and most recommender systems in general, is that they do not model properly a human-to-human interaction, where the user and the adviser interact. Focusing on this problem, *conversational recommender system* have been recently proposed (e.g., [35], [85]). These systems provide dialogues supporting the customer in the selection process. Thus, by adding user feedbacks, the system can make a better idea of the type of product the user may be interested in. This kind of recommender systems is out of scope of this work.

This Section presents an approach that, as the SVD-based approaches, uses the information from all the users, not only from the correlated ones. Notice that the *prediction* task in recommender systems can be seen as a way of filling in the missing entries in the data matrix. As in Section 6.3, an adaptation of Alternation technique is proposed as imputation method. In this case, it is adapted to the problem of recommender systems. Hence, missing rates are predicted and the dimension of the data is reduced. One of the advantages of the Alternation technique over the SVD is that, since it can deal with missing data, it is not necessary to fill in the missing rates with zeros or averages before applying it. On the other hand, its computational cost is not as high as in the case of the SVD. The proposed approach is focused on the *prediction* (not *recommendation*) and its performance is compared with one of the method proposed in [99].

This Section is structured as follows. First of all, the approach proposed in [99] is briefly introduced. Then, the proposed adaptation of the Alternation technique to the problem of prediction task in recommender systems is presented. Data sets and experimental results are provided. Finally, concluding remarks are summarized.

6.4.1 A SVD-based Approach: Sarwar et al.'s Proposal

Since the proposed adaptation of Alternation technique to recommender systems is compared later to the Sarwar et al.'s [99] approach, this Section presents briefly their proposal.

For the shake of simplicity, it will be referred to as Sarwar's approach hereinafter. As mentioned above, this approach is based on the SVD. Hence, first of all, given a matrix of rates W that contains missing data, its missing entries must be filled in.

Sarwar et al. [99] propose to fill in the missing data with the corresponding column average. Then, the entries are normalized by subtracting the corresponding row average. Once the data matrix W has not missing entries, the SVD is computed, given the decomposition: $W = U\Sigma V$. The best rank-r approximation to W is obtained by keeping only the r largest singular values of S, giving Σ_r. Accordingly, the dimensions of U and V are also reduced and the matrix: $W_r = U_r \Sigma_r V_r^t$ is the closest rank-r matrix to W.

Finally, the products of the factors $U_r \Sigma_r^{1/2}$ and $\Sigma_r^{1/2} V_r^t$ are used to obtain the predictions. Concretely, the predicted rate $W(i,j)$ of the customer i to the product j is obtained with the following expression:

$$W(i,j) = \overline{w}_j + (U_r \Sigma_r^{1/2})(i)(\Sigma_r^{1/2} V_r^t)(j) \tag{6.35}$$

where $\overline{w}_j$ is the jth-column average.

One disadvantage of this approach is that the computational cost is very high. Due to that fact, with concrete data sets, the number of customers and products must be reduced. In this thesis, this approach is implemented by using the aforementioned *thin* SVD, in order to reduce its computational cost.

6.4.2 Adaptation of the Alternation Technique to Recommender Sytems

This Section proposes a variant of the Alternation technique, adapted to the *prediction* task in the recommender systems.

One fact that should be highlighted is that the distribution of missing data in this particular problem is random, which in general allows to obtain good results with the Alternation, even with high percentages of missing data.

Given a matrix of rates $W_{c\times p}$, the goal of the Alternation technique is to find the factors $A_{c\times r}$ and $B_{r\times p}$ such that minimizes the expression $\|W_{c\times p} - A_{c\times r}B_{r\times p}\|_F^2$, where r is the rank of the data matrix. The product AB is the best rank-r approximation of the matrix W in the sense of the Frobenius norm [43]. Although the aim is the same as the one of the SVD, the Alternation allows to deal with missing data and additionally, it has a far smaller computational cost. As in the case of the SVD, A and B are r dimensional representations of customers and products, respectively.

Recall that the Alternation technique is a two-step algorithm, which starts with one random factor (A_0 or B_0) and compute one factor at a time, until the product of the factors AB converge to W. A filled matrix $W_{imputed}$ is obtained with the product of these factors: $W_{imputed} = AB$. Focusing on the recommender systems, it can be used the fact that the entries in W take values in a known interval: $[m, M]$, where m and M are the minimum and the maximum rate values, respectively. The idea is to enforce that the range of values of the entries in $W_{imputed}$ lies in this known interval. This is tackled later.

First of all, at each step of the Alternation algorithm, the rows of A (B respectively) are normalized. The proposed variant of the Alternation is summarized in the following steps:

Algorithm: Given a data matrix $W_{c\times p}$, which contains the rates given by different customers:

1. Take a random matrix A_0, normalize its rows.

2. Compute B_k from A_{k-1}:

$$B_k = (A_{k-1}^t A_{k-1})^{-1} A_{k-1}^t W \tag{6.36}$$

3. Normalize the rows of B_k.
4. Compute A_k from B_{k-1}:

$$A_k = W B_k^t (B_k B_k^t)^{-1} \tag{6.37}$$

5. Normalize the rows of A_k.
6. Repeat the steps 2-5 until the product $W_{imputed} = A_k B_k$ converges to W.

Solution: W_{impute} contains the prediction rates values.

Notice that the classical Alternation algorithm consists in applying repeatedly the steps 2 and 4 from above until the convergence is achieved.

Hence, with the normalization steps added to the algorithm, each entry of the imputed matrix $W_{imputed}(i,j)$ is the scalar product between the *ith*–unitary row of A and the *jth*–unitary column of B. That is:

$$W_{imputed}(i,j) = \mathbf{a}_i \cdot \mathbf{b}^j = \cos(\alpha_{ij}) \tag{6.38}$$

where $\mathbf{a}_i$ and $\mathbf{b}^j$ are the *ith*–unitary row of A and *jth*–unitary column of B, respectively and α_{ij} could be interpreted as the angle between them (if they are considered as vectors).

Therefore, if no restrictions are added, the entries of $W_{imputed}$ would take values in the interval $[-1,1]$. However, as mentioned above, the aim is to achieve that the values in $W_{imputed}$ lie in the interval $[m,M]$. Hence, the values of the entries in $W_{imputed}$ should be transformed. One possibility is to transform them directly by using a linear transformation. The problem is that the *cosine* function is not linear. Instead of the *cosine*, the angles are studied. This means that the *arccosine* must be applied to obtain the angles. Therefore, an interval where the *cosine* is invertible should be defined: the initial punctuation ratings must be transformed in order to take values in the interval $[0,1]$ instead of $[-1,1]$.

In addition, and in order to weight equally all the possible values, instead of considering values in $[m,M]$, the interval $[m-0.5, M+0.5]$ is taken (otherwise, the values m and M are not considered as the rest of values).

The following steps summarize the transformation that gives the initial values from the interval $[m,M]$ to $[0,1]$:

1. $\widetilde{W} = \frac{(W-m)}{\Delta}$, where $\Delta = M - m$, $\widetilde{W}(i,j) \in [0,1]$

2. $\widetilde{W} = \pi\widetilde{W}$, $\widetilde{W}(i,j) \in [0,\pi]$

3. $\widetilde{W} = \cos(\widetilde{W})$, $\widetilde{W}(i,j) \in [0,1]$

The Alternation technique is applied to the transformed matrix $\widetilde{W}$ and a filled matrix of prediction rates $\widetilde{W}_{imputed}$ is obtained. Finally, the above transformations must be undone:

$$W_{imputed} = \frac{\arccos(\widetilde{W}_{imputed})}{\pi}\Delta + m \tag{6.39}$$

Hence, the values in $W_{imputed}$ lie in $[m, M]$, as the values in the initial missing data matrix W. Notice that the *arccosine* can be applied because it is invertible in the interval $[0, \pi]$.

6.4.3 The r Selection

The rank used to obtain the best predicted rates depends on the approach used in the prediction task.

Unfortunately, the rank of the matrix is not known a priori and it can not be directly computed, when working with missing data. Hence, the problem of estimating the rank of a missing data matrix W should be faced out. It is shown in Chapter 4 that in the single object case in the SFM problem, the rank of the matrix of trajectories is at most 4. Recall that it is due to the dimensions and the nature of the factors obtained in the decomposition: the Motion and Shape matrices, both of rank 4 at most, are recovered. In the particular problem of recommender systems, the obtained factors have not a similar interpretation.

Sarwar et al. [99] point out that it is important to choose the optimal r in order to obtain good predicted values. They search for a r-value large enough to capture all the important information in the matrix, while small enough to avoid overfitting errors. However, the results do not vary so much considering different rank values.

In the incremental SVD presented by Brand [15], it is proposed to use the rank value as a measure of complexity of the model. The objective is to maximize the probability of correct generalization, while minimizing the complexity of the model. Furthermore, Brand points out that users ratings data have poor repeatability from day to day. Therefore, a good low-rank approximation of the data has higher probability of generalization than a medium-rank model that perfectly reconstructs the data. His experiments show that the incremental SVD, with rank 4 or 5, predicts the missing rates better than matrices with higher rank. Brand points out that the higher the singular values are, the more constrained the imputation is by previous inputs, and therefore, the better the estimated SVD. With only a few rated items, the SVD has small singular values. In general, in those cases, a smaller rank will give better predicted rates.

It is shown in Chapter 5 that, if W has rank r and the Alternation technique is used to approximate it by a rank $\tilde{r}$ matrix, being $\tilde{r} > r$, noise is added to the data during the process, in order to achieve a higher rank matrix. Consequently, the missing entries are wrongly filled in. On the contrary, if the matrix is approximated by a rank-$\tilde{r}$ matrix, with $\tilde{r} < r$, information is lost during the process. The missing entries are again wrongly filled in. Hence, the goodness of the predicted values obtained with the Alternation depends on the rank value used. In Section 6.4.5, it will be shown that using the proposed adaptation of the Alternation for recommender systems, the best predicted rates are obtained, in general, for $r = 4$ or $r = 5$, as in [15]. In extreme cases, with large amount of missing data, $r = 3$ or $r = 2$ are enough.

Having in mind the importance of a proper rank selection, our experimental results and comparisons have been performed considering a range of different rank values. The error considering each rank value is computed. Finally, the smallest rank for which a minimum error is obtained is selected. Actually, a similar procedure is carried out in the aforementioned approaches. The case $r = 1$ is not considered, for any approach, since it makes no sense to project the data onto a 1-dimensional subspace.

6.4.4 Data Sets

Data sets used in the experiments are introduced in this Section. Concretely, three different public data sets are considered.

The first data set is the one provide by the MovieLens recommender system (http://www.movielens.umn.edu), which is a Web-based research recommender system. One of the data sets they give consists of 100,000 ratings (discrete values from 1 up to 5) from 943 users and 1,682 movies. A user-movie matrix W, formed by 943 rows and 1,682 columns, is constructed. Each entry $W(i, j)$ represents the rating (from 1 up to 5) of the ith-user on the jth-movie. This data set is also used in [99] and [15]. Since the goal is to study the goodness of the obtained predicted values, some entries are randomly removed and used to study their recovered values. These entries form the test data set. The rest of the entries used to recover the data are the train data set. Concretely, five different train and test data sets, split into 80,000 train and 20,000 test cases, are also given at: http://www.movielens.umn.edu. The initial data matrix has 93.69% of missing data, while using any train data set, a matrix of 94.95% of missing data is obtained.

The second data set is the one obtained from BookCrossing, a service where book lovers exchange books all around the world and share their experiences with others (http://www.bookcrossing.com). Ziegler et al. [135] collect data from 278,858 members of BookCrossing, referring to 271,379 different books. A total of 1,157,112 rates are provided. These rates take implicit (0) and explicit (from 1 up to 10) discrete values. The data set used in [135] is available at: http://www.informatik.uni-freiburg.de/∼cziegler. Ziegler at al. pointed out that in many recommender systems, the suggested items are similar to the ones rated by the user. For instance, in some cases, the obtained list is very similar respect to author, genre or topic (if the items are books). This kind of lists are not very useful to the users, since they do not give new information to them. Ziegler et al. [135] propose the *topic diversification* in order to increase the diversity of a top-N list recommended products. The idea is to study the user's extent of interest in specific topics. It is proposed a new *intra-list similarity metric*, which measure the diversity in the given recommended list. If all available data is used, the obtained data matrix is extremely sparse (concretely, it has a percentage of missing data of about 99.9968%). In our experiments, a smaller matrix with a higher density of known data, is considered. In particular, it is required that each user rates a minimum of books. At the same time, only the books that have been rated by a minimum of users are considered. Different minimum values will be considered in the experiments, as presented in Section 6.4.5.

Finally, the last data set used in the experiments is obtained from Jester, an online joke recommender system: http://eigentaste.berkeley.edu. The complete data set is publicly available at: http://www.ieor.berkeley.edu/∼goldberg/jester-data/ . Concretely, 4.1 million continuous rating (from -10 up to 10) of 100 jokes from 73,421 users are provided. In this case, different users (rows) are selected randomly, given a matrix with smaller dimensions. The data set is presented in [42], where Goldberg et al. present a collaborative filtering algorithm based on the principal component analysis (PCA) to obtain the predictions in the Jester recommender system. In particular, the authors propose to project the data into the eigenplane with the PCA. Then, the projected data are clustered by using a recursive rectangular clustering. When a new user ask for recommendations, first, the rates the user gives are projected onto the eigen plane. Then, the representative cluster of the user is found. Finally, the recommendations are computed from the rates collected in the cluster.

6.4.5 Experimental Results

As mentioned above, this work is focused on the *prediction* task in recommender systems. The performance of our approach is compared with the one of the method presented in [99]. For the comparison, the Mean Absolute Error (MAE) is used as a measure of goodness of the recovered values. This is the measure of goodness used in most proposed approaches for recommender systems and it is defined as follows:

$$MAE = \frac{1}{N}\sum_{i,j}|P_{ij} - W_{ij}| \tag{6.40}$$

where i, j correspond to the indexes of the artificially removed entries in W (test data set), N is the number of these removed entries and P_{ij} is the obtained predicted value for the entry W_{ij}.

Due to the different nature of the data in each data set, different experiments are carried out with each one and the obtained results are presented separately in every data set. For instance, the percentage of available data and the size o the matrices are different in each data set. Another characteristic that should be taken into account is that the rates can take discrete or continuous values.

MovieLens Data Sets

In this data set, rates take integer values from 1 up to 5 and the percentage of missing data is about 94.05%. With such amount of missing data, experiments considering different percentages of missing data would not give significant conclusions. Five different train/test sets are used in the experiments and the mean of all the train/test sets are given.

Testing different rank values

Different r-dimension values (equivalently, r-rank values) are tested (from 2 up to 20) and the one for which the MAE is minimum is chosen.

In Fig. 6.28, the mean of the obtained results in the five test/train data sets, for each rank value, is plotted. The minimum error (MAE) obtained with the proposed approach (denoted as ALT in the plots) is smaller than the one obtained with the approach proposed in [99] (denoted as Sarwar's approach and as SVD in the plots), as it can be seen in Fig. 6.28 (top-left). Concretely, with the Alternation-adapted approach, the smallest MAE value is obtained in the rank-4 case and its value is 0.7703. With Sarwar's approach, the minimum error ($MAE = 0.7772$) is obtained in the rank-16 case. The results presented in [99] are a little different: the obtained MAE is about 0.7400 for the rank-14 case. This difference is possibly due to different test/train splits. Notice that, in both approaches, a MAE with this value means an error of about ± 1 in the prediction task. As pointed out in [15], this is very accurate, since difference may reach ± 2 values if the user is asked in different days.

From our results, it can be concluded that the data space can be reduced to a 4-dimensional subspace, as it is also concluded in [15], where this data set is also studied. Unfortunately, open source of that algorithm is not available for a fair comparison. Sarwar et al. [99], found that a 14-dimensional subspace is needed in order to capture all the variance in the data. In fact, in their approach, the MAE does not change much considering different dimensions, as it is shown in Fig. 6.28 (top-left).

The standard deviation σ (STD) of the error is also computed (Fig. 6.28 (top-right)). With our approach and for the rank-4 case $\sigma = 0.9796$, while with Sarwar's approach $\sigma = 0.9858$, for the rank-16 case.

The computational cost for both approaches is depicted in Fig. 6.28 (bottom-left). Our method is clearly faster, for all the tested rank values.

Although the initially known values will not be modified, since only the predicted values are required, in Fig. 6.28 (bottom-right), and only as an illustration, it is shown the obtained rms error computed by taking the train data set. That is, by considering only the initially known data. Using our proposed approach, the rms takes values below 0.5, while the rank value is higher than 2. Hence, the initially known rates would not be changed with our proposed technique if the results would be rounded to the nearest integer. In the case of Sarwar's approach, the rms takes values higher than 0.8 for any rank value. In this case, the rates would be modified if the results would be rounded to the nearest integer.

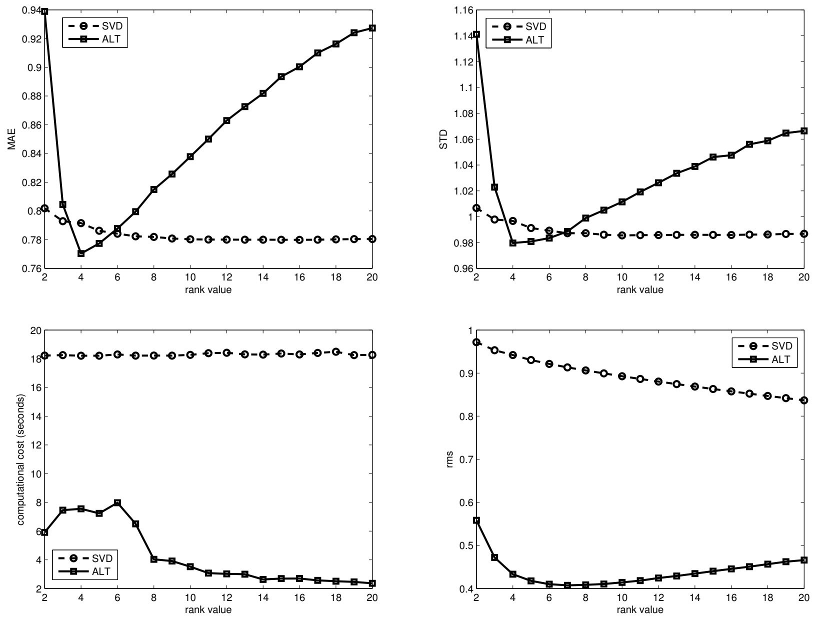


Figure 6.28: performance study, testing different r values.

Imposing a minimum of known entries in rows/columns

Although other approaches do not take it into account, it makes sense to impose a minimum of known elements in each row/column of the data matrix. In this second experiment, a minimum number of movies rated by each customer and also a minimum number of customers that rate each movie are required.

A study of the obtained results imposing a different number of minimum known entries in each row/column (n hereinafter) is presented in Fig. 6.29. The previous experiment corresponds to the $n = 0$ case. For every n, different rank values are considered and the one for which the MAE is minimum is plotted. In particular, the rank that gives the minimum MAE is in general 4 or 5 with our approach, while it takes different values (higher than 5) in the case of Sarwar's approach (Fig. 6.29 (bottom-left)).

For all studied parameters, results are better as n increases. This is due to the fact that, as n increases, both the size of the studied data matrix and the percentage of missing data decrease.

It can be seen in Fig. 6.29 (top-left), that the MAE decreases as n increases for both approaches and it is smaller with our approach. Concretely, the plot corresponds to the rank value that gives the minimum MAE. Fig. 6.29 (top-right) shows that the STD value decreases as n increases for both approaches. It takes smaller values with the Sarwar's approach, in general.

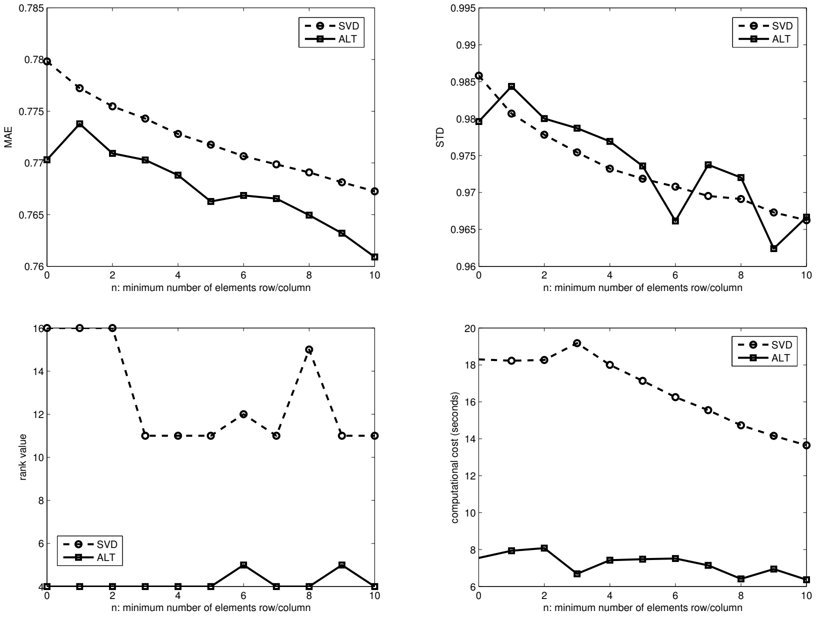


Figure 6.29: performance study, considering different n values (minimum known elements per row and column).

The computational cost of both approaches is shown in Fig. 6.29 (bottom-right). Notice that it is always smaller with our approach and it decreases as n increases in both approaches.

Rounding results

Finally, and specially for applications with integer rates values, the obtained predictions could be rounded. In this case, rounding the values obtained in $W_{imputed}$, they take discrete values $\{1, \cdots, 5\}$.

The obtained results, which are analogous to the ones of the previous experiment (plotted in Fig. 6.29), are showed in Fig. 6.30. With our approach, the minimum *MAE* is achieved again in the case of rank-4 as before and its value is 0.7390. With Sarwar's approach, the minimum is obtained for the rank-17 case and its value is 0.7458.

BookCrossing Data Set

The matrix obtained with this data set is very sparse. Concretely, the matrix has a percentage of missing data of about 99.9968%. In order to obtain a matrix with more density of data, the users that have rated less than 20 books are discarded, while at the same time, only the books that have been rated by at least 200 users are considered. Since books and users are discarded at the same time, the above

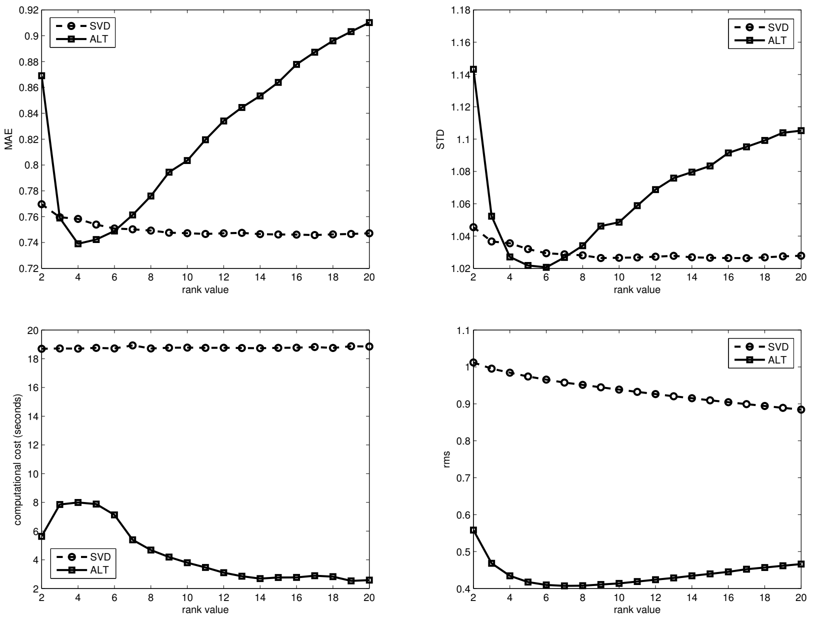


Figure 6.30: performance study by rounding the obtained predictions, case $n = 0$.

conditions do not means that every row and column have more than 20 and 200 known entries, respectively. In fact, with this two conditions, the number of considered users (rows) and books (columns) are 17,197 and 193, respectively and the percentage of missing data is about 98.5094%. Again, 5 different test/train data sets are generated. Concretely, the test data set contains 0.4906% of known data, while the train data set only 1% of known data.

Fig. 6.31 (left) shows the obtained MAE values, considering different rank values. It can be seen that the minimum error is obtained with the Alternation, for $r = 2$. Concretely, $MAE = 3.5811$. With Sarwar's approach, the minimum error is also achieved for $r = 2$ and $MAE = 3.7535$. With only a percentage of 1% of known data, a 2-rank matrix predicts better the missing rates than a matrix with a higher rank.

Notice that the error in this case is larger than in the experiments with the MovieLens data set. However, since the ratings lie in different ranges, another measurement should be defined. Goldberg et al. [42] propose to use the *Normalized Mean Absolute Error* ($NMAE$) in order to compare errors obtained from different data sets. The $NMAE$ is defined as:

$$NMAE = \frac{MAE}{M - m} \tag{6.41}$$

where M and m are the maximum and minimum value in the range of rates, respectively. Using this new measure of error, the results obtained with the proposed approach and both data sets can be compared: in the case of the MovieLens data set, $MAE = 0.7704$, which gives $NMAE = 0.1926$, while in the case of BookCrossing

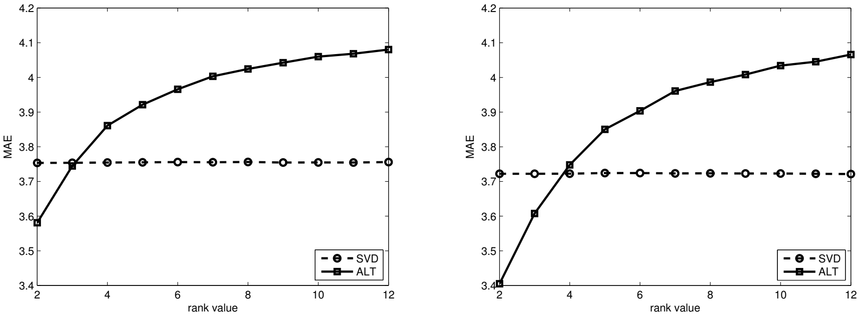


Figure 6.31: obtained MAE considering different rank values; (left) the data matrix has 98.5094% of missing data; (right) the data matrix has 99.0465% of missing data.

data set, the MAE obtained is 3.5811, which gives $NMAE = 0.3581$. Effectively, the error with this second data set is higher than with the MovieLens' one. Recall that with this second data set, the percentage of missing data is higher.

In order to test different data matrices, a similar experiment is carried out, requiring a smaller number of rates per book. Concretely, books from the original matrix that have been rated by less than 150 users are discarded. With these two conditions, the number of considered users and books are 21,026 and 354, respectively and the percentage of missing data is about 99.0465%. Hence, the obtained matrix has larger dimensions and more missing data. Concretely, test data set contains 0.1535% of data, while the train data set contains only 0.8% of data.

The obtained MAE values in this case are plotted in Fig. 6.31 (right). Notice that the minimum MAE is obtained with the Alternation, for $r = 2$. Its value is 3.4051. With the Sarwar's approach, the $MAE = 3.7215$ and it is obtained for $r = 12$. However, similar results are obtained for any rank value.

Jester Data Sets

The rates in this data set take continuous values from -10 up to 10. The obtained matrix contains the rates given by 73,421 users to 100 jokes and the percentage of known data is about 48%.

Testing different percentages of missing data

Different percentages of missing data are generated by randomly removing data (concretely, 60%, 70%, 80% and 90%). The removed entries form the test data set, while the rest of the data form the train data set. Again, 5 different train/test data sets are considered at each experiment. Only 18,000 users from the total of 73,421 are randomly selected for the experiments, as in [42].

Fig. 6.32 shows the error (MAE) value obtained considering different percentages of missing data and different rank values (from 2 up to 20). It can be seen that in the case of Sarwar's approach, the obtained MAE is quite similar for any rank value. Concretely, the minimum MAE is achieved with $r = 14$ and $r = 12$, with 60% and 70% of missing data. With percentages of missing data of about 80% and 90%, the minimum MAE is obtained with $r = 3$ and $r = 2$, respectively. In the case of the Alternation, the MAE value does depend on the rank value, as it can be appreciated in Fig. 6.32. The minimum MAE is obtained for $r = 5$, while the percentage of missing data is below 90%, in which case, the minimum MAE is obtained with $r = 4$.

The minimum MAE obtained with each percentage of missing data is plotted in Fig. 6.33, considering the rank that gives the minimum value. It is shown that, the minimum MAE is achieved with the Alternation approach, for any percentage of missing data.

The obtained MAE values are similar to the ones presented in [42], which studies the same data set. Unfortunately, no comparison can be performed with [42], since the authors do not provide an accurate information about the percentage of missing data they consider, nor the used rows (they select 18,000 rows randomly).

Although the obtained MAE seems to be higher than the obtained with the MovieLens data set, the ratings lie in different ranges. If the $NMAE$ proposed by Goldberg et al. [42] is used, it can be seen that the results are similar with both data sets. In the case of the MovieLens data set, the values obtained with the proposed approach are as follows: $MAE = 0.7704$, which gives $NMAE = 0.1926$, while in the case of Jester data set, the MAE obtained with 90% percentage of missing data is 3.8959, which gives $NMAE = 0.1948$. Therefore, the goodness of the predicted rates is quite similar in both cases.

6.4.6 Conclusions

The Alternation technique is adapted to tackle the *prediction* task in recommender systems. Concretely, a variant of this technique, which uses the fact that the rates take values in a known interval, is presented. As in the SVD-based approaches, not only the correlated customers are used in the *prediction* task, but also the non correlated ones.

The variant of the Alternation is compared to the one presented in [99]. Three different public data sets, obtained from three different recommender systems, are studied. Experimental results show that the proposed approach performs better than the SVD used in [99], both regarding the error value and also the computational cost. It can be concluded that the data space can be reduced to a low-dimensional subspace, in most cases, with the proposed approach.

It should be highlighted the good results obtained with the Alternation, even with percentages of missing data of more than 90%. This is due to the fact that the missing data in the data matrix present a random distribution.

The iterative multiresolution scheme is not used, since the idea in this application is to use all the available data and not only the local one. It would have no sense to

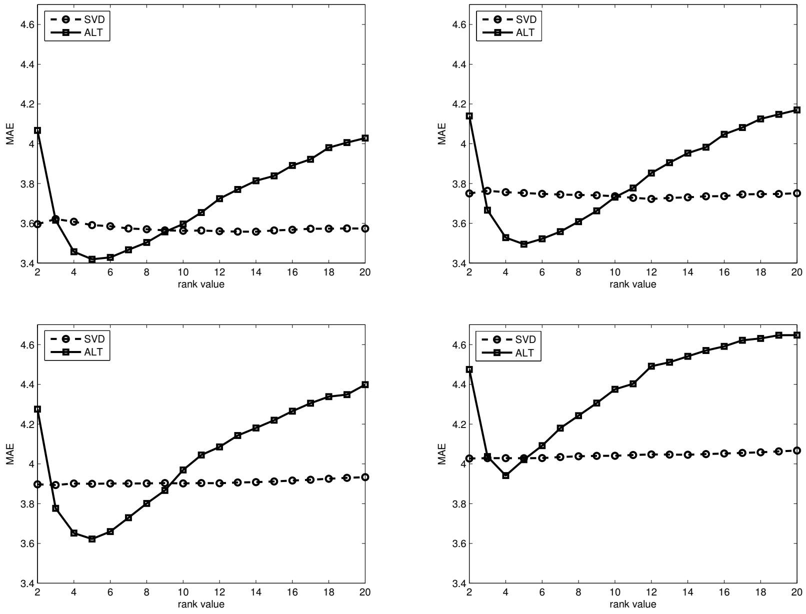


Figure 6.32: obtained *MAE* values for different rank values and different percentages of missing data; (top) 60% and 70%; (bottom) 80% and 90%.

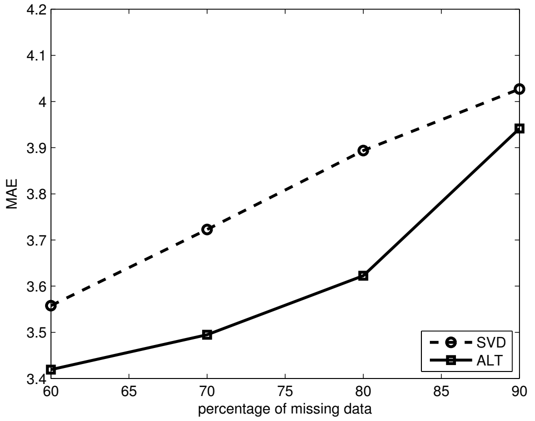


Figure 6.33: obtained *MAE* values for different percentages of missing data, considering the rank value that gives the minimum *MAE*.

search for prediction values considering rates from a few users, not even users that have similar tastes.

6.5 Summary

This Chapter shows different applications in which the Alternation technique can be applied. One of the proposed applications is in the computer vision framework, while the other two are in the bioinformatics (DNA microarray data) and marketing (recommender systems) contexts.

For each application, an introduction is provided in order to introduce the context. An adaptation of the Alternation technique is presented to tackle the particular problem. Finally, experimental results are given in order to show the good performance of each Alternation-adaptation proposed. Real data are used in all the cases.

As a conclusion, although the Alternation technique has been considered in this thesis to tackle the SFM problem, it is shown that it can be useful in other applications.

Chapter 7

Summary and Contributions

This Chapter summarizes the main contributions and conclusions exposed in previous Chapters. Future lines of research are also included here.

7.1 Summary

The Structure from Motion (SFM) problem consists in extracting the 3D object shape as well as the relative camera-object motion from 2D coordinates of tracked features. In computer vision context, **factorization technique** is a method that can address this problem. The idea of the factorization is to decompose the matrix of trajectories, which contains 2D coordinates of tracked feature points, into the motion and shape matrices. The motion matrix contains the relative camera-object motion, while the shape matrix contains the 3D coordinates of the feature points. These matrices can be computed by using the fact that the matrix of trajectories has a reduced rank. Concretely, in the case of a single rigid moving object, the rank is at most 4. Hence, the matrix of trajectories can be approximated by a rank-4 matrix.

If the trajectories are full, the SVD gives the best rank-4 matrix approximation of the matrix of trajectories. The problem is that if the matrix of trajectories contains missing data, the SVD can not be applied. Several techniques have been proposed to deal with missing data, but a closed solution has not been found yet. Actually, some of these proposed techniques can not deal with high percentages of missing data. This thesis is focused on the missing data matrix factorization to tackle the SFM problem. Concretely, the Alternation technique, whose properties and algorithm are presented, is used through the thesis. A detailed State of the Art of factorization techniques focused on the Structure from Motion problem is reported.

One of the goals of this thesis is to present a method that allows to deal with high percentages of missing data. The idea is to fill in (partially or totally) missing data in the input matrix and apply a factorization technique to this filled in matrix instead to the input one, which has a high percentage of missing data.

In the multiple objects case with missing data, the factorization technique can not be directly applied to recover the shape and motion of each object. This is due to the fact that trajectories are not sorted into different objects. The idea is to compute the motion segmentation of the trajectories and apply a single object technique to obtain the shape and motion of each of the objects. However, another problem should be faced out: the estimation of the rank of the matrix of trajectories. It can not be directly computed by using the SVD, since there are missing data. Furthermore, it is difficult to estimate without using any prior information about the scene, such as number of objects or kind of motion. Different rank values are considered and a factorization technique is used to fill in missing data. A measure of goodness should be defined in order to choose the best rank value. Therefore, missing data would be filled in with the factorization technique and, at the same time, rank would be estimated. Once a full matrix has been obtained, a motion segmentation approach could be applied and the SFM of each object could be computed.

This thesis is centered on the SFM problem through factorization. However, the idea is to adapt this technique, in particular the Alternation technique, to different applications, not only in the computer vision context. Several problems have been analyzed, searching for similarities and differences with the SFM problem. For each studied problem, an adaptation of the Alternation technique is proposed to tackle it. Concretely, the studied problems are: photometric stereo, missing data in DNA microarrays (in the bioinformatics context) and prediction task in recommender systems.

7.2 Contributions

This thesis proposes different approaches concerning the SFM with missing data for the single and multiple objects cases. Furthermore, adaptations of the Alternation technique to several applications, apart from the SFM, have also been presented. The main contributions of this thesis are summarized below.

7.2.1 Alternation Technique

- **Different missing data distributions** are presented and the performance of the Alternation with each distribution is compared. Concretely, a random and non-random distribution are studied. It is shown that the Alternation performs better with a random distributions of the missing data.

- **The filtering capability of the Alternation** technique is also studied. Experimental results show that, in general, the Alternation filters the noise in the data. Only in cases with a high percentage of missing data or when working with a very small matrix, the Alternation can not filter the noise.

7.2.2 SFM in the Single Object Case with Missing Data

- It is presented a **variant of the Alternation technique**: the ***Alternation with motion constraints***. It consists in imposing the orthonormality of the camera axes at each iteration of the algorithm. The advantage of this variant of Alternation is that an Euclidean reconstruction is obtained. Furthermore, the possibility of adding prior knowledge to the problem is proposed. In particular, the **smooth camera motion** can be imposed, by fixing the angle between camera axes of consecutive frames.

- In order to deal with matrices that contain a high percentage of missing data, **an iterative multiresolution scheme** is proposed. The objective is to fill in (partially or totally) the input matrix considering submatrices with a low percentage of missing data. The Alternation technique is applied to each of these submatrices, filling in missing data with the product of the obtained factors. The final goal is to improve the results when a factorization technique is applied to the matrix filled in with the proposed scheme, instead of applying it directly to the input matrix, which has a high percentage of missing data. An evaluation study of the proposed scheme is carried out. It is shown that, when the percentage of missing data is high, a factorization technique applied to the matrix filled in with the proposed scheme gives better results than when it is applied directly to the input matrix. However, when the percentage of missing data is low, the factorization technique performs quite well applied to the input matrix directly.

- The most common **measure of goodness** to study the performance of a factorization technique is the root mean square error *rms*, which studies how well the initially known data are approximated. This measure does not take into account how the missing data are filled in. If all the entries in the input matrix are initially known, the $\mathbf{rms_{all}}$, which studies how the missing data are recovered, is also computed. This is not a realistic situation, but it allows to study the goodness of the recovered data. Furthermore, in the experiments with synthetic data in which the shape and motion ground truth are known, the $\mathbf{rms_M}$ and $\mathbf{rms_S}$ are also computed. These errors measure the goodness of the recovered motion and shape matrices, respectively.

7.2.3 SFM in the Multiple Objects Case with Missing Data

- **A novel technique for rank estimation of a missing data matrix of trajectories**, corresponding to multiple rigid moving objects, is proposed. The underlying idea of the proposed technique is that, since feature points belong to rigid objects, the behaviour of the recovered missing entries due to the movement should be similar to the one of the initially known entries. If this movement is interpreted as a signal, the movement similarity is identified with the fact that the **frequency spectra** of the input matrix should be preserved after recovering missing data. The proposed technique consists in considering different rank values and applying the Alternation technique to fill in missing data. A measure of goodness, which compares the frequency spectra of each filled in matrix with the one of the input matrix, is defined. The good performance of the proposed technique considering sequences with both, synthetic and real data, is empirically shown. Concretely, from the sequences with synthetic data it can be concluded that the rank of the input matrix is well estimated, even with high percentages of missing data. Real data experiments show that the proposed technique is able to deal with noisy data.

- A well-known motion segmentation approach is implemented and the proposed **rank estimation approach is tested**. Sequences with synthetic data, which contain several objects (from two up to nine), are generated. Additionally, real sequences from a publicly available data set are also tested. A comparative of the performance of the motion segmentation approach, considering different rank values, is carried out. It is shown that the obtained error in the clustering is, in general, smaller when the rank is estimated with the proposed technique. Hence, the **significance of a correct rank estimation** before applying a motion segmentation approach is shown.

- Once the trajectories have been segmented into different objects, a single object factorization technique is applied, in order to obtain the 3D shape and the relative camera-object motion of each of the objects. Furthermore, examples of the recovered factors, obtained when there are some errors in the segmentation, are provided. It is shown that **results become worse as the number of wrongly clustered features increases.**

7.2.4 Applications

- The **Alternation technique is adapted to tackle the photometric stereo problem**. The goal is to obtain the surface normals and reflectance of an object from a given set of images obtained under varying illumination. The grey intensities of the images are stacked into a *measurement matrix*. The fact that the set of images of a Lambertian object can be approximated by a low-dimensional linear subspace is used. Concretely, if the images do not have shadows nor specularities, the *measurement matrix* can be approximated by a rank-3 matrix. However, in real images there are usually shadows, specularities and an ambient background illumination. Therefore, a rank-4 formulation, which includes an ambient term, is introduced. **The entries corresponding to shadows and specularities are considered as missing entries**. Hence, they do not affect the final result. The proposed Alternation-based approach is used to fill in those missing entries. Experimental results show improvements when those entries are considered as missing data.

- In a different application, **Alternation is adapted in order to fill in missing entries in DNA microarray gene expression data**. The main drawback in this particular application is that the data matrix has a full rank. That is, the rank is equal to the number of columns (which is in general a small number).

Hence, with only a few missing data, the Alternation can not give as good results as expected. Due to that fact, an Alternation-based strategy is proposed. A comparative of the Alternation-based strategy with several approaches presented in the literature is provided. It is shown that **results obtained with Alternation are similar to the ones of the best methods proposed in this field**. Furthermore, no parameter needs to be estimated with the Alternation technique. Although, in general, the Alternation-based strategy does not outperforms those methods, a novel field where Alternation can be used is proposed.

- **The Alternation is adapted to be used in recommender systems**. The idea is to predict the rate values that a customer would give to different items, using all information stored in the system. In particular, this information is provided by the current customer and also by all the customers that have used the system. Different rank values are considered and, the one for which the obtained error is smaller, is chosen. In general, the best prediction rates are achieved with a rank value of 4 or 5. If the percentages of missing data is very high, a rank of 2 or 3 is enough. Experimental results with real data sets, used in real recommender systems, show the good **performance of the proposed approach** compared to a technique based on SVD.

7.3 Future Lines of Research

Throughout this dissertation, several future lines of research have been identified. Some of them are summarized below.

- **A closed form solution** for the case of missing data has not been found yet. In the full data case, the SVD gives a closed solution to the SFM problem through factorization.
- More **elaborated camera models** could be tested (e.g., perspective, or weak-perspective). The affine camera model is used through the thesis.

In the single objects case, when trying to add priors:

- Constraints about the smoothness of the camera motion could be added to the formulation by using a **non-linear approach**, instead of imposing a concrete increment of the rotation angle between frames.

With respect to the multiple objects case:

- An analytic demonstration of the **convergence of the proposed rank estimation technique** could be studied in order to support this approach.

Finally, concerning the applications:

- **The factors obtained with the Alternation-based strategy in DNA microarray could be studied**. Some techniques based on SVD use these factors to find clusters of genes, which can set useful relationships between genes.

- Analogously, **factors obtained with the Alternation-adapted to recommender systems could be exploited**. They could be very useful in marketing applications. For instance, to obtain clusters of similar customers or of similar products.

- **Other applications** could be studied by using the corresponding Alternation-adapted technique.

Bibliography

[1] H. Aanaes and R. Fisker. Robust factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1215–1225, 2002.

[2] P.M.Q. Aguiar and J.M.F. Moura. Weighted factorization. In *IEEE International Conference on Image Processing*, pages 549–552, 2000.

[3] P.M.Q. Aguiar and J.M.F. Moura. Three-dimensional modeling from two-dimensional video. *IEEE Transactions on Image Processing*, 10:1541–551, 2001.

[4] P.M.Q. Aguiar and J.M.F. Moura. Rank 1 weighted factorization for 3D structure recovery: algorithms and performance analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1134–1149, 2003.

[5] O. Alter, P.O Brown, and D. Botstein. Processing and modeling genome-wide expression data using singular value decomposition. In *Microarrays: Optical Technologies and Informatics, SPIE*, volume 4266, pages 171–186, 2001.

[6] P. Anandan and M. Irani. Factorization with uncertainty. *International Journal of Computer Vision (IJCV)*, 49:101–116, 2002.

[7] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:218–233, 2003.

[8] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72:239–257, 2007.

[9] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions? In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 270–277, 1993.

[10] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. In *Society for industrial and applied mathematics (SIAM)*, volume 37, pages 573–595, 1995.

[11] D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54, 1998.

[12] T.E. Boult and L.G. Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, pages 179–186, 1991.

[13] M. Brand. Morphable 3D models from video. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 493–500, 2001.

[14] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *ECCV*, pages 707–720, 2002.

[15] M. Brand. Fast online SVD revisions for lightweight recommender systems. In *SIAM International Conference on Data Mining*, 2003.

[16] M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 122–128, 2005.

[17] M. Brand. Fast low-rank modifications on the thin singular value decomposition. *Linear Algebra and its Applications*, 415:20, 2006.

[18] S. Brandt. Closed-form solutions for affine reconstruction under missing data. In *Proceedings Statistical Methods for Video Processing Workshop, in conjunction with ECCV*, pages 109–114, 2002.

[19] A. Brazma, A. Robinson, G. Cameron, and M. Ashburner. One-stop shop for microarray data. *Nature*, 403:699–700, 2000.

[20] C. Bregler, A. Hertzmann, and N. Biermann. Recovering non-rigid 3D shape from image streams. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[21] P. Brown and D. Botstein. Exploring the new world of the genome with DNA microarrays. *Nature Genetics*, 21:33–37, 1999.

[22] A. Buchanan. Investigation into matrix factorization when elements are unknown. *Technical report*, 2004.

[23] A. Buchanan and A.W. Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 316–322, University of Oxford, 2005.

[24] P. Chen and D. Suter. Recovering the missing components in a large noisy low-rank matrix: Application to SFM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1051–1063, 2004.

[25] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iteration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1098–1104, 1996.

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

[27] F. De la Torre and M.J. Black. Robust principal component analysis for computer vision. In *IEEE International Conference on Computer Vision*, pages 362–369, 2001.

[28] A. Del Bue. Extracting average shapes from occluded non-rigid motion. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 2 of *Lecture Notes on Computer Science*, pages 218–225, Girona, June 2007.

[29] A. Del Bue and L. Agapito. Non-rigid 3D shape recovery using stereo factorization. In *Asian Conference on Computer Vision (ACCV)*, volume 1, pages 25–30, 2004.

[30] A. Del Bue, X. Lladó, and L. Agapito. Segmentation of rigid motion from non-rigid 2D trajectories. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 1 of *Lecture Notes on Computer Science*, pages 491–498, Girona, June 2007.

[31] J.L. DeRisi, V.R. Iyer, and P.O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.

[32] M.B. Eisen and P. Brown. DNA arrays for analysis of gene expression. *Enzymol*, 303:179–205, 1999.

[33] R. Epstein, A.L. Yuille, and P.N. Belhumeur. Learning object representations from lighting variations. In *Object recognition workshop, ECCV*, 1996.

[34] H. Farid and J. Kosecká. Estimating planar surface orientation using bispectral analysis. *IEEE Transactions on image processing*, 16:2154–2160, 2007.

[35] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 11:11–34, Winter 2006-2007.

[36] M. Fischler and R. Bolles. Random sample consensus: A paradigm fo model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24:381–395, 1981.

[37] S. Friedland, A. Niknejad, and L. Chihara. A simultaneous reconstruction of missing data in DNA microarrays. *Linear Algebra and its Applications*, 416:8–28, 2006.

[38] S. Friedland, A. Niknejad, M. Kaveh, and H. Zare. An algorithm for missing value estimation for DNA microarray data. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 1092–1095, 2006.

[39] C.W. Gear. Feature grouping in moving objects. In *IEEE Workshop on motion of non-rigid and articulated objects*, pages 214–219, 1994.

[40] J.L. Gerton, R. DeRisi, J.L. Shroff, M. Lichten, P.O. Brown, and T.D. Petes. Global mapping of meiotic recombination hotspots and coldspots in the yeast Saccharomyces cerevisiae. In *Proceedings of the national academy of sciences (PNAS)*, volume 97, pages 11383–11390, 2000.

[41] A. Goh and R. Vidal. Segmenting motions of different types by unserpervised manifold clustering. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[42] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. In *Information Retrieval*, volume 4, pages 133–151, 2001.

[43] G.H. Golub and C.F. Van Loan, editors. *Matrix Computations*. The Johns Hopkins Univ. Press, 1989.

[44] B.B. Gonçalves and P.M.Q. Aguiar. Complete 3-D models from video: a global approach. In *IEEE International Conference on Image Processing ICIP'04*, volume 4, pages 2479–2482, Singapore, 2004.

[45] A. Gruber and Y. Weiss. Factorization with uncertainty and missing data: Exploiting temporal coherence. In *Advances in Neural Information Processing Systems 16*. 2004.

[46] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 707–714, 2004.

[47] A. Gruber and Y. Weiss. Incorporating non-motion cues into 3D motion segmentation. *Computer Vision and Image Understanding (CVIU)*, 108:271–271, 2007.

[48] R.F.C. Guerreiro and P.M.Q. Aguiar. Factorization with missing data for 3D structure recovery. In *IEEE Workshop on multimedia signal processing*, pages 105–108, 2002.

[49] R.F.C. Guerreiro and P.M.Q. Aguiar. 3D structure from video streams with partially overlapping images. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 897–900, 2002.

[50] R.F.C. Guerreiro and P.M.Q. Aguiar. Estimation of rank deficient matrices from partial observations: two-step iterative algorithms. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 450–466, 2003.

[51] N. Guilbert, A. Bartoli, and A. Heyden. Affine approximation for direct batch recovery of euclidian structure and motion from sparse data. *International Journal of Computer Vision (IJCV)*, 69:317–333, 2006.

[52] A. Guitton. Huber solver versus IRLS algorithm for quasi L1 inversion. Technical Report 103, Standford exploration project, April 2000.

[53] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. *International Journal of Computer Vision (IJCV)*, 53:285–300, 2000.

[54] R. Hartley and F. Schaffalitzky. Powerfactorization: 3D reconstruction with missing or uncertain data. In *Australian-Japan advanced workshop on Computer Vision*, 2003.

[55] R. Hartley and A. Zisserman, editors. *Multiple view geometry in computer vision.* Cambridge University Press, Book web page at: http://www.robots.ox.ac.uk/˜vgg/hzbook/index.html, accessed Febrary 2007, 2000.

[56] H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *Optical Society of America*, 11:3079–3089, 1994.

[57] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17:981–991, 1999.

[58] http://www.auburn.edu/academic/agriculture/fisheries/genomics/. A review of DNA microarray data analysis. Accessed December 2007.

[59] J. Hu, H. Li, M. S. Waterman, and X.J. Zhou. Integrative missing value estimation for microarray data. *BMC Bioinformatics*, 7, 2006.

[60] D.Q. Huynh, R. Hartley, and A. Heyden. Outlier correction in image sequences for the affine camera. In *IEEE ICCV*, volume 1, pages 585–590, 2003.

[61] D.Q. Huynh and A. Heyden. Outlier detection in video sequences under affine projection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 695–701, 2001.

[62] D.Q. Huynh and A. Heyden. Robust factorization for the affine camera: analysis and comparison. In *International Conference on Control, Automation, Robotics and Vision Computing (ICARVC)*, volume 1, pages 126–131, 2002.

[63] N. Ichimura. Motion segmentation based on factorization method and discriminant criterion. In *ICCV: International Conference on Computer Vision*, volume 1, pages 600–605, 1999.

[64] N. Ichimura and F. Tomita. Motion segmentation based on feature selection from shape matrix. In *Systems and Computers in Japan*, volume 31, pages 32–42, 2000.

[65] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV*, volume 1, pages 626–633, 1999.

[66] D.W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer vision and image understanding (CVIU)*, (1):57–81, 2001.

[67] H. Jia, J. Fortuna, and A. Martinez. Perturbation estimation of the subspaces for structure from motion with noisy and missing data. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 1101–1007, 2006.

[68] C. Julià, A. Sappa, F. Lumbreras, J. Serrat, and A. López. Factorization with missing and noisy data. In *Computational Science ICCS 2006: 6th International Conference*, volume 1 of *LNCS*, pages 555 – 562, Reading, UK, 2006.

[69] C. Julià, A. Sappa, F. Lumbreras, J. Serrat, and A. López. An iterative multiresolution scheme for SFM. In *ICIAR 2006, LNCS 4141*, volume 1, pages 804–815, 2006.

[70] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *International Jounal of Computer Vision*, 33:163–180, 1999.

[71] K. Kanatani. Statistical optimization and geometric inference in computer vision. *Philosophical transactions: Mathematical, physical and engineering sciences*, 356:1303–1320, 1998.

[72] K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 586–591, 2001.

[73] K. Kanatani. Model selection for geometric inference. In *ACCV: The 5th Asian Conference on Computer Vision*, volume 1, pages 21–32, 2002.

[74] K. Kanatani and C. Matsunaga. Estimating the number of independent motions for multibody motion segmentation. In *ACCV: The 5th Asian Conference on Computer Vision*, 2002.

[75] K. Kanatani and Y. Sugaya. Factorization without factorization: complete recipe. *Memoirs of the Faculty of Engineering, Okayama University*, 38:61–72, 2004.

[76] H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21:187–198, 2005.

[77] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 22:1495–1502, 2007.

[78] T. Kim and K.S. Hong. Estimating approximate average shape and motion of deforming objects with a monocular view. *International Journal of Pattern Recognition and Artificial Intelligence*, 19:585–601, 2005.

[79] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[80] Y. Li and M. Brooks. An efficient recursive factorization method for determining structure from motion. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages –143, 1999.

[81] X. Lladó, A. Del Bue, and L. Agapito. Euclidean reconstruction of deformable structure using a perspective camera with varying intrinsic parameters. In *International Conference on Pattern Recognition (ICPR06)*, 2006.

[82] Y. Ma, J. Soatto, J. Koseck, and S Sastry. *An Invitation to 3D Vision: From Images to Geometric Models.* Springer-Verlang, New York, 2004.

[83] D. Martinec and T. Pajdla. Structure from many perspective images with occlusions. In *European Conference on Computer Vision (ECCV)*, 2002.

[84] D. Martinec and T. Pajdla. 3D reconstruction by fitting low-rank matrices with missing data. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 198–205, 2005.

[85] R. McGinty and B. Smyth. Adaptive selection: an analysis of critiquing and preference-based feedback in conversational recommender systems. *International Journal of Electronic Commerce*, 11:35–57, Winter 2006-2007.

[86] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):858–867, 1997.

[87] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *International Conference on Computer Vision*, pages 696–702, 1998.

[88] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19:2088–2096, 2003.

[89] T. Okatani and K. Deguchi. On the Wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72:329–337, 2007.

[90] A. Pascual-Montano, P. Carmona-Sez, R.D. Pascual-Marqui, F. Tirado, and J.M. Carazo. Two-way clustering of gene expression profiles by sparse matrix factorization. In *IEEE Computational Systems Bioinformatics Conference Workshops (CSBW'05)*, 2005.

[91] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:206–218, 1997.

[92] M.H. Pryor. The effects of Singular Value Decomposition on collaborative filtering. *Computer Science Technical Report*, 1998.

[93] L. Quan and T. Kanade. A factorization method for affine structure from line correspondences. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 1996.

[94] P. Resnick, N. Iacovou, M. Suchak, M. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Computer Supported Cooperative Work (CSCW)*, 1994.

[95] S. Roweis. EM algorithms for PCA and SPCA. In *Neural Information Processing Systems (NIPS)*, pages 626–632, 1997.

[96] S. Roweis and L. Saul. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[97] A. Ruhe and A. Wedin. Algorithms for separable nonlinear least squares problems. *Technical report*, 1974.

[98] A. Ruhe and A. Wedin. Algorithms for separable nonlinear least squares problems. *Society for Industrial and Applied Mathematics (SIAM)*, 22:318–337, 1980.

[99] M.S. Sarwar, G. Karypis, J.A. Konstan, and J.T. Riedl. Application of dimensionality reduction in recommender system - a case study. In *E-Commerce Workshop*, pages 309–324, 2000.

[100] J.A. Scales and A. Gersztenkorn. Robust methods in inverse theory: Inverse problems. Technical Report 4, 1988.

[101] I. Scheel, M. Aldrin, I.K. Glad, R. Sorum, H. Lyng, and A. Frigessi. The influence of missing value imputation on detection of differentially expressed genes from microarray data. *Bioinformatics*, 21:4272–4279, 2005.

[102] A. Shashua. On photometric issues in 3D visual recognition from a single 2D image. *International Journal of Computer Vision*, 21:99–122, 1997.

[103] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligent*, 22:888–905, 2000.

[104] H. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):854–867, 1995.

[105] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the cell*, 9:3273–3297, 1998.

[106] P. Sturm and W. Triggs. A factorization based algorithm for multi-image projective structure from motion. In *European Conference on Computer Vision (ECCV)*, pages 709–720, 1996.

[107] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Workshop on Statistical Methods in Video Processing, LNCS*, volume 3247, pages 13–25, 2004.

[108] K. Sungshik. A novel recovery algorithm of incomplete observation matrix for converting 2D video to 3D content. In *International Workshop on Intelligent Computing in Pattern Analysis/Synthesis (IWICPAS), LNCS 4153*, pages 260–269, 2006.

[109] E. Tola, S. Knorr, E. Imre, A. Alatan, and T. Sikora. Structure from motion in dynamic scenes with multiple motions. In *Workshop On Immersive Communication And Broadcast Systems (ICOB)*, 2005.

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

[111] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 493–500, 2001.

[112] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1110–1115, 2005.

[113] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 845–851, 1996.

[114] B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15:617–625, 1997.

[115] R. Tron and R. Vidal. A benchmark for the comparison of 3D motion segmentation algorithms. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[116] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:1–6, 2001.

[117] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and GPCA. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 310–316, 2004.

[118] R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 510–517, 2004.

[119] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[120] G. Wang and J. Wu. Deformation weight constraint and 3D reconstruction of nonrigid objects. In *ICIAR, LNCS 4633*, pages 285–294, 2007.

[121] H. Wang and N. Ahuja. Facial expression decomposition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 958–965, 2003.

[122] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *ICCV: International Conference on Computer Vision*, volume 2, pages 975–982, 1999.

[123] T. Wiberg. Computation of principal components when data is missing. In *Second Symposium of Computational Statistics*, pages 229–326, 1976.

[124] J. Xiao, J. Chai, and T. Kande. A closed-form solution to non-rigid shape and motion recovery. In *European Conference on Computer Vision (ECCV)*, 2004.

[125] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 815–821, 2005.

[126] J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision (ECCV)*, pages 94–106, 2006.

[127] J. Yuan and E. Oja. Projective nonnegative matrix factorization for image compression and feature extraction. In *ACIA, LNCS 3540*, pages 333–342, 2005.

[128] A.L. Yuille, D. Snow, R. Epstein, and P.N. Belhumeur. Determining generative models of objects under vaying illumination: shape and albedo from multiple images using SVD and integrability. *International Journal of Computer Vision*, 35:203–222, 1992.

[129] L. Zelnik-Manor and M. Irani. Multiview constraints on homographies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:214–223, 2002.

[130] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 287–293, 2003.

[131] L. Zelnik-Manor and M. Irani. Temporal factorization vs. spatial factorization. In *ECCV*, volume 2, pages 434–445, 2004.

[132] L. Zelnik-Manor, M. Machline, and M. Irani. Multi-body segmentation: revisiting motion consistency. In *VMODS Workshop in conjunction with ECCV*, 2002.

[133] L. Zhang, C. Brian, A. Hertzmann, and S.M. Seitz. Shape and motion under varying illumination: unifying structure from motion, photometric stereo, and multi-view stereo. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 618–625, 2003.

[134] H. Zhou and T.S. Huang. Recovering articulated motion with a hierarchical factorization method. In *Gesture Workshop*, pages 140–151, 2003.

[135] C. Ziegler, S.M. McNee, J.A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *International World Wide Web Conference (WWW '05)*, 2005.

publications

- C. Julià, Motion segmentation trough factorization. Application to night driving assistance, *Master Thesis, Computer Vision Center, Universitat Autònoma de Barcelona, September 2004*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Motion segmentation through factorization. Application to night driving assitance, *VISAPP, Volume 2, 2006.*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Factorization with missing and noisy data, *Computational Science ICCS 2006: 6th International Conference, 2006, LNCS 3991.*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, An Iterative Multiresolution Scheme for SFM, *ICIAR 2006, LNCS 4141-1, pp. 804-815, 2006.*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, An iterative scheme for matrix factorization with missing data, *Computer Vision: Progress of Research and Development. 1st CVC Internal Workshop, 2006.*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Motion segmentation from feature trajectories with missing data, *3rd Iberian Conference on Pattern Recognition and Image Analysis, 2007, LNCS 4477-1.*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, An approach for motion segmentation with missing data, *Computer Vision: Progress of Research and Development. 2nd CVC Internal Workshop, 2007.*
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Rank estimation in 3D multibody motion segmentation, *Electronics Letters,* (submitted September 2007, accepted January 2008)
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, An iterative multiresolution scheme for SFM with missing data, *Journal of Mathematical Imaging and Vision,* (under review, November 2007)
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Recovery of surface normals and reflectance from different lighting conditions, *ICIAR 2008* (under review, January 2008)

- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Photometric stereo through an adapted Alternation approach, *ICIP 2008,* (under review, January 2008)
- C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Rank estimation in missing data matrix problems, *IEEE Transaction on Image Processing,* (submitted, January 2008)