Image and Vision Computing 28 (2010) 164–176

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

An iterative multiresolution scheme for SFM with missing data: Single and multiple object scenes

Carme Julià [a,*], Angel D. Sappa [b], Felipe Lumbreras [b,c], Joan Serrat [b,c], Antonio López [b,c]

[a] Intelligent Robotics and Computer Vision Group, Department of Computer Science and Mathematics, ETSE-URV Av.Països Catalans 26, 43007 Tarragona, Spain
[b] Computer Vision Center, Edifici O Campus UAB, 08193 Bellaterra, Barcelona, Spain
[c] Computer Science, Department Edifici O Campus UAB, 08193 Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:
Received 23 July 2007
Accepted 26 May 2009

Keywords:
Structure from Motion
Factorization techniques

ABSTRACT

Most of the techniques proposed for tackling the *Structure from Motion* problem (SFM) cannot deal with high percentages of missing data in the matrix of trajectories. Furthermore, an additional problem should be faced up when working with multiple object scenes: the rank of the matrix of trajectories should be estimated. This paper presents an iterative multiresolution scheme for SFM with missing data to be used in both the single and multiple object cases. The proposed scheme aims at recovering missing entries in the original input matrix. The objective is to improve the results by applying a factorization technique to the partially or totally filled in matrix instead of to the original input one. Experimental results obtained with synthetic and real data sequences, containing single and multiple objects, are presented to show the viability of the proposed approach.

1. Introduction

The *Structure from Motion* problem (SFM) consists in extracting the 3D shape of a scene as well as the camera 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 [26] many variants have been presented in the literature (e.g., [24] for the case of paraperspective camera model; a sequential factorization method in Refs. [21,5,10] for the multiple object case, etc.). The 2D image coordinates of a set of 3D features (points in general, but also lines [25] and planes [22]) 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:

$$W_{2f\times p} = M_{2f\times r}S_{r\times p} \tag{1}$$

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 [7]. These factors can be estimated thanks to the key fact that W is rank deficient and due to constraints derived from the orthonormality of the camera axes.

The *Singular Value Decomposition* (SVD) [7] 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.

1.1. Related work

In their seminal paper, Tomasi and Kanade [26] propose an initialization method in which they first decompose the largest full sub-matrix by the factorization method and then the initial solution grows by one row or by one column at a time, filling in the missing data. The main drawback of this technique is that finding the largest full sub-matrix is a NP-hard problem. Jacobs [12] 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. One drawback of this approach is that the solution is strongly affected by noise on the data. It is used as an initialization by other approaches. An iterative algorithm for recovering missing components in a large noisy low-rank matrix is provided by Chen and Suter [4]. The algorithm begins with a complete sub-matrix 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 capacity* versus missing data in order to decide which parts of the matrix

* Corresponding author. Tel.: +34 93 581 38 06; fax: +34 93 581 16 70.
E-mail addresses: cjulia@cvc.uab.es, cjuliaferre@yahoo.com (C. Julià).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.05.015

should be used in the iterative process. The goal is to recover the most *reliable* incomplete sub-matrix by using the iterative algorithm. Then, other columns and rows are projected on it using an imputation method. In Ref. [13], Jia et al. present an algorithm that aims the SFM recovery with noisy and missing data. It is similar to the aforementioned one [12], but instead of selecting several r-tuple of columns, it uses the most reliable sub-matrix to recover the 3D structure. The authors define a criterion that provides a measure of the sensitivity of a sub-matrix to perturbation due to noise: the *deviation parameter*. Using this criterion, the sub-matrices with smallest deviation parameter are considered to build the final matrix.

Wiberg [29] 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 into two sets and compute them alternatively. In a recent paper, Okatani and Deguchi [23] present in detail Wiberg algorithm focusing on the matrix factorization problem and demonstrating its good performance compared to the Levengerg–Marquardt (LM) technique.

Wiberg's algorithm is generally referenced in the literature (e.g., [3,8]), as the origin of what is called the *Alternation* technique. This iterative technique starts with an initial random A_0 or B_0 and, at each iteration k, computes alternatively each of the factors A_k and B_k, until the product A_kB_k converges to W. The key point of this 2-step algorithm is that, since the updates of A given B (analogously in the case of B given A) can be independently done for each row of A (or column of B), missing entries in W correspond to omitted equations. Due to that fact, the method could fail to converge when the amount of missing data is large. Several variants of this approach have been proposed in the literature. In Ref. [8], Guerreiro and Aguiar introduce the *Row-Column* algorithm, which is very similar to the Alternation technique. They study its performance and compare it with the Expectation-Maximization (EM) algorithm. They conclude that it performs better than the EM and, besides, it is more robust to the initialization. Hartley and Schaffalitzky [11] suggest to add a normalization step between the two factor updates at each iteration. This particular Alternation technique is denoted as PowerFactorization. Furthermore, the authors propose another variant to Alternation, focussing on the SFM problem. In this case, A and B factors correspond to the motion M and shape S matrices, respectively. Hence, since S contains the 3D feature points in homogeneous coordinates, it can be imposed that the last row of S is equal to $\mathbf{1}$ (where $\mathbf{1}$ represents a vector of 1). In Ref. [1], Aanaes and Fisker present a factorization scheme, based on the Alternation, that can deal with mismatched features, missing features and noise on the features. In Ref. [3], Buchanan and Fitzgibbon summarize factorization approaches with missing data and propose the *Alternation/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.

Additionally, several techniques that are not purely factorization have been proposed to tackle the SFM problem with missing data. Martinec and Pajdla [20] propose a technique for 3D reconstruction by fitting low-rank matrices with missing data. 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. This method does not try to fill in the missing data in the matrix of trajectories. In fact, only the known data are used. The formulation is similar to the one presented in Ref. [12]. The main difference is that the problem is formulated in terms of the original subspaces, while in Ref. [12] the complementary ones are used. Finally, Guilbert et al. [9] present a batch method for recovering an Euclidian structure and motion from sparse image data. Using closure constraints [27], the camera coefficients are formulated linearly in the entries of the affine fundamental matrices.

In summary, most of the proposed SFM approaches tackle only the single object case. In the multiple object case, trajectories belonging to the same object are first clustered together. Then, the structure and motion of each object in the scene are recovered by applying a single object SFM technique.

1.2. Objective

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. Addressing to this problem, an iterative multiresolution scheme, which fill in missing data in the matrix of trajectories was presented in Ref. [14]. Improvements to this approach were presented in Ref. [15] and, more recently, in Ref. [17]. The key point of this approach is to work with sub-matrices, 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, missing entries in each selected set can be filled in just by multiplying the recovered factors obtained by applying a factorization technique. Experimental results in Ref. [17] study the performance of the proposed iterative multiresolution scheme by considering different factorization techniques. Concretely, results obtained with the Alternation [29], the PowerFactorization [11], and the Alternation/Damped Newton [3] are compared. It is shown that the Alternation technique is the most appropriate to fill in missing data by using this iterative multiresolution scheme. This paper is an extension of the iterative multiresolution scheme presented in Ref. [17] to the case of multiple objects in the scene.

It is well known that the rank of the matrix of trajectories corresponding to a single rigid object is at most four [5]. Unfortunately, in the multiple object case the rank of the matrix of trajectories is not known a priori. Actually, it is not even bounded when the number of objects in the scene is not known. Therefore, an additional problem should be faced up in the multiple object case: the rank of the matrix of trajectories should be estimated before applying any factorization technique. The current paper uses the missing data matrix rank estimation technique proposed in Ref. [16], as explained in Section 2.2.

The proposed approach should be seen as a pre-processing technique; that is, first 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 fill in missing entries of the whole matrix of trajectories. In the single object case, the recovered factors correspond to the motion and structure of the whole matrix. In the multiple object case, trajectories belonging to the same object should be first clustered together; then, the structure and motion of each of the objects in the scene can be independently computed. 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 paper is organized as follows. Section 2 presents the extension of the iterative multiresolution scheme to the multiple object case. Experimental results considering synthetic and real data sequences, containing single and multiple objects, are reported in Section 3. Section 4 contains conclusions and future work.

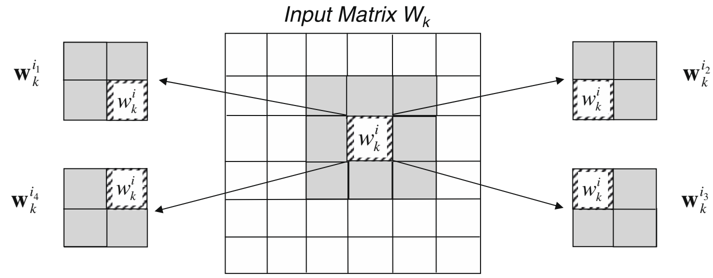

Fig. 1. w_k^i and its four corresponding $\mathbf{w}_k^{i_n}$ matrices, computed during the first stage, at iteration $k = 6$.

2. Iterative multiresolution scheme: single and multiple object cases

Essentially, the basic idea of the iterative multiresolution scheme is to work with sub-matrices that have a reduced percentage of missing data. Then, a factorization technique is used to decompose each sub-matrix into two factors A and B and the missing data are filled in with the resulting product AB. The proposed approach consists of two stages, which are described below.

2.1. Observation matrix splitting

Let $W_{2f\times p}$ be the given missing data matrix of trajectories (also referred to through the paper as original input matrix) of p feature points tracked over f frames. For the sake of presentation simplicity, hereinafter denoted as W. Feature points belong to a scene containing a single or several rigid objects. Let k be the index indicating the current iteration number.

The aim at this first stage is to split the matrix of trajectories W up into sub-matrices 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 up into a set of $k \times k$ non-overlapped sub-matrices, defined as w_k^i, with a size of $\lfloor\frac{2f}{k}\rfloor \times \lfloor\frac{p}{k}\rfloor$[1] and where $i = 1, \ldots, k^2$. 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 (sub-matrix 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 sub-matrices $\mathbf{w}_k^{i_n}, n = 1, \ldots, 4$ (see Fig. 1) with twice the size of w_k^i, and for every sub-matrix w_k^i.

The idea of this enlargement process is to study the recovered entries in w_k^i when different size overlapped regions are considered together. The latter avoids obtaining results from small single regions. Other strategies were tested in order to compute in a fast and robust way sub-matrices with a lower percentage of missing entries (e.g., quadtrees, ternary graph structure), 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. 2 illustrates the five partitions of matrix W_k—i.e., the one generated by all the w_k^i and the other four ones, obtained with all the $\mathbf{w}_k^{i_n}$ sub-matrices, generated at the sixth iteration. When all these matrices are considered together, the overlap between the different areas is obtained, see textured cell in Figs. 1 and 2.

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 $\mathbf{w}_k^{i_n}$).

2.2. Sub-matrices processing

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

Different factorization techniques could be used as imputation method at this stage. The performance of this iterative multiresolution scheme considering different factorization techniques is studied in Ref. [17]. In particular, the Alternation [29], the PowerFactorization [11] and the Alternation/Damped Newton are tested. Reported results in Ref. [17] show that the Alternation is the most appropriate method for the iterative multiresolution scheme.

As mentioned in Section 1, an additional problem should be faced up in the multiple object case: the rank of the matrix of trajectories should be estimated before applying any factorization technique. Missing data matrix rank estimation is out of scope of the current paper. It has been a stand-alone research topic during the last decades in the computer vision community (e.g., [2,18,30]), as well as other research fields (e.g., [6,28]). In the current work, the technique proposed in Ref. [16] is used to estimate the rank of the given input matrix, while the percentage of missing data is lower than 20%. Then, this rank value is used through the whole experiments (matrices containing a high percentage of missing data) to process both, originally given input matrices as well as the corresponding sub-matrices. 3 is intended to compare results from the proposed iterative scheme with respect to those obtained from the Alternation, but not how they are affected by a wrong rank value.

Independently of their size hereinafter sub-matrices will be referred to as W_s. Therefore, given a sub-matrix W_s with a percentage of known data of at least 50%, its corresponding A_s and B_s matrices are obtained by using the Alternation technique, taking the previously estimated rank r. Then, the product A_sB_s is used to fill in the matrix W_s. Finally, the root mean squared (rms) is computed as follows:

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

[1] $\lfloor\frac{a}{b}\rfloor$ correspond to the integer part of the quotient $\frac{a}{b}$.

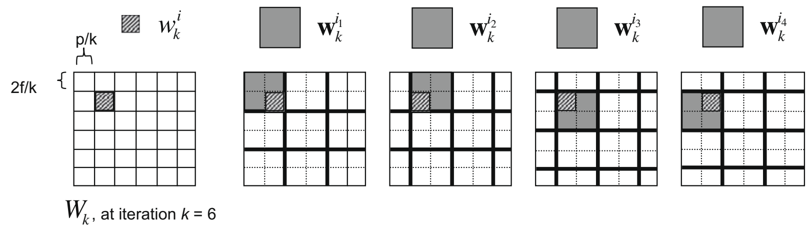

Fig. 2. Five partitions of matrix W_k. Note the overlap between a w_k^i sub-matrix with its corresponding four $\mathbf{w}_k^{i_n}$ sub-matrices, computed during the first stage.

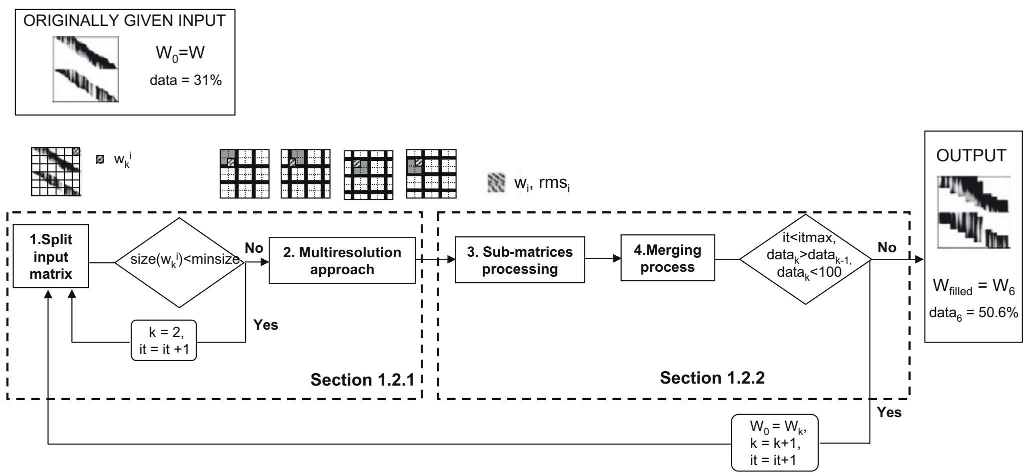

Fig. 3. Algorithm. Example of an input matrix W with only 31% of known data. The filled matrix W_{filled}, obtained after six iterations, contains 50.6% of known data.

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 in W_s is associated with a weight, defined as $\frac{1}{rms_s}$.

Finally, when every sub-matrix 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 sub-matrix (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 2.1) splitting the new matrix W up either by increasing k by one or, in case the size of sub-matrices 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 full. An outline of the algorithm can be found below; Fig. 3 presents an overview of the algorithm.

2.2.1. Outline of the algorithm

Inputs: W, original input matrix of trajectories; r, rank of matrix[2] W; *data*, percentage of known data in W; *itmax*, maximum number of iterations; *minsize*, sub-matrix 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 up into $k \times k$ sub-matrices 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 $\mathbf{w}_k^{i_n}, n = 1, \ldots, 4$).
(3) Sub-matrices processing: apply Alternation to all the sub-matrices, considering the estimated rank value r.
(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: $W_{filled} = W_k, data_k > data_0$.

3. Experimental results

This Section presents an evaluation of the performance of the iterative multiresolution scheme in the single and multiple object cases. The aim is to study the robustness to missing and noisy data of a factorization technique applied to the partially or totally filled in 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, 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}.

Taking into account results obtained in Ref. [17], the Alternation technique is used inside the proposed scheme. Actually, it is also

[2] r is estimated by using [16] while the percentage of missing data is lower than 20%.

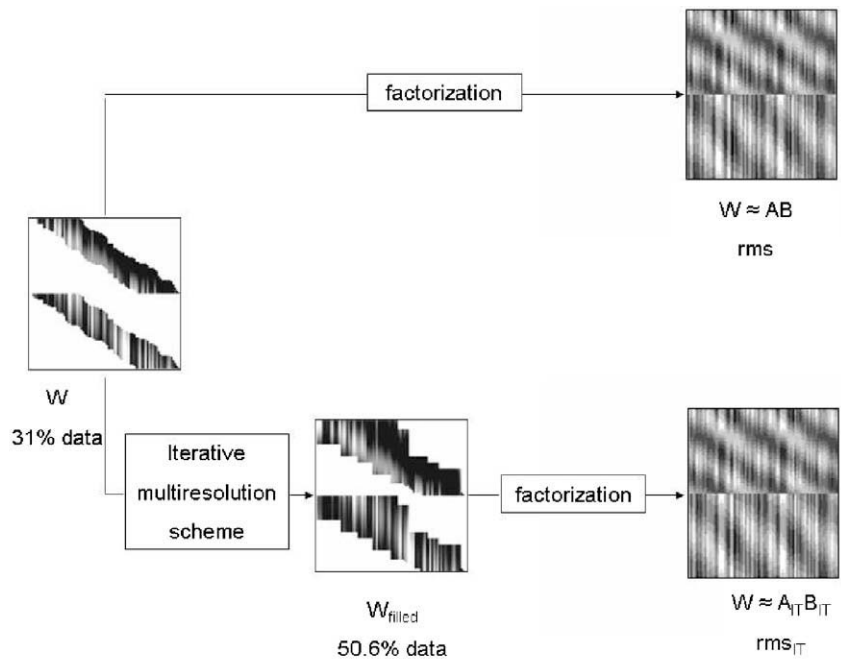

Fig. 4. Evaluation study: comparison of *rms* obtained in each case. In our experiments, the Alternation is used as factorization technique. IT stands for iterative scheme. In this example, the input matrix only contains 31% of known data.

used in the evaluation study to split the whole matrix of trajectories up into shape and motion matrices.

Experiments using both synthetic and real data are presented below. Different amounts of missing data are considered—from 10% up to 70%. Furthermore, in the synthetic case, different levels of Gaussian noise are added to the 2D feature point trajectories—standard deviation σ with a value from $\frac{1}{3}$ to 1 and zero mean. Only results corresponding to the case of $\sigma = 1$ are shown in the current paper, just to illustrate the case of noise in the data. 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:

$$rms = \frac{\|\widehat{W} - AB\|_F}{\sqrt{n}} = \sqrt{\frac{\sum_{i,j}|(\widehat{W})_{ij} - (AB)_{ij}|^2}{n}} \tag{3}$$

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

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 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. 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. Besides, the performance of factorization techniques are far better dealing with random missing data and it may not be appropriate for an evaluation study.

As pointed out in Ref. [4], the *rms* defined by the expression (3) 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:

$$rms_{all} = \frac{\|W_{all} - AB\|_F}{\sqrt{2fp}} \tag{4}$$

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

3.1. Synthetic data

This Section presents 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. 5(left). The second data set is generated from 3D points of a triangular mesh (nodes), representing a Beethoven sculptured surface, see Fig. 9(left). In this second object the points are not as uniformly distributed as in the previous object. Different sequences are obtained with these objects by performing rotations and translations 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.

3.1.1. Single object case

The first sequence containing a single object is defined by 200 frames containing 300 features from the first data set. The trajectories are plotted in Fig. 5(right). Fig. 6 shows an example of recovered shape (left) and motion ((middle) and (right)) obtained by applying the Alternation technique to the matrix filled in with the proposed iterative scheme. The original input matrix has about 20% of missing data.

The obtained *rms* considering different percentages of missing data are shown in Fig. 7. Concretely, a *boxplot* representation is used in order to show the *rms* obtained in the 100 attempts. These boxplots enclose data in between the lower and the upper quartiles

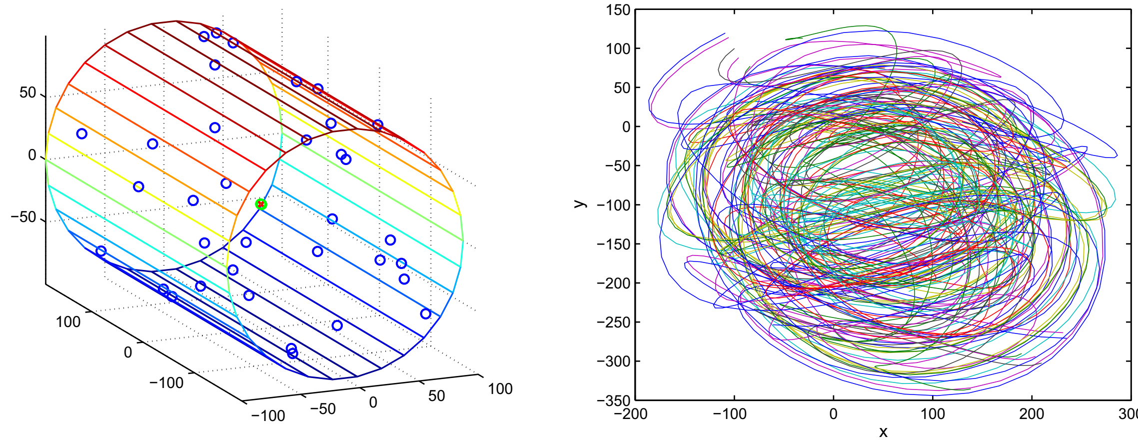

Fig. 5. Synthetic object: (left) cylinder; (right) feature point trajectories represented in the image plane.

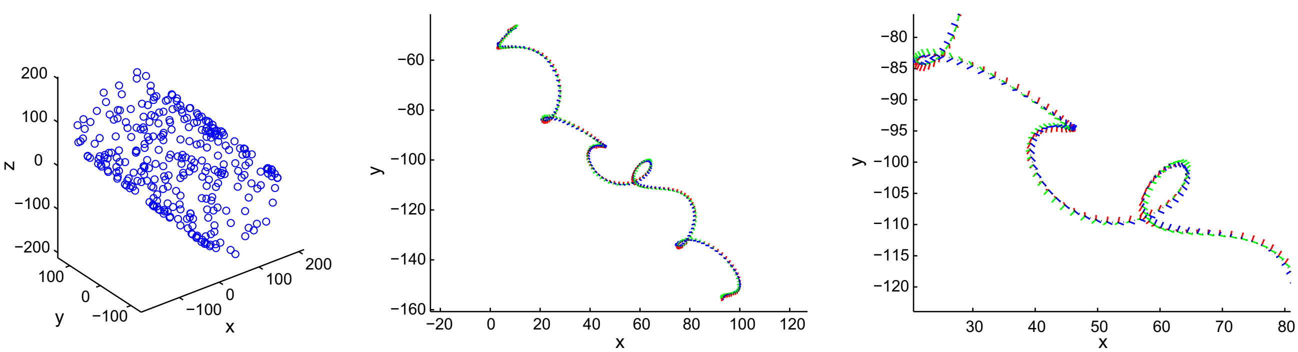

Fig. 6. Cylinder scene: (left) 3D reconstruction: notice that there are thicker points which correspond to the reappearing features; (middle) recovered camera motion; (right) zoom in the recovered camera motion.

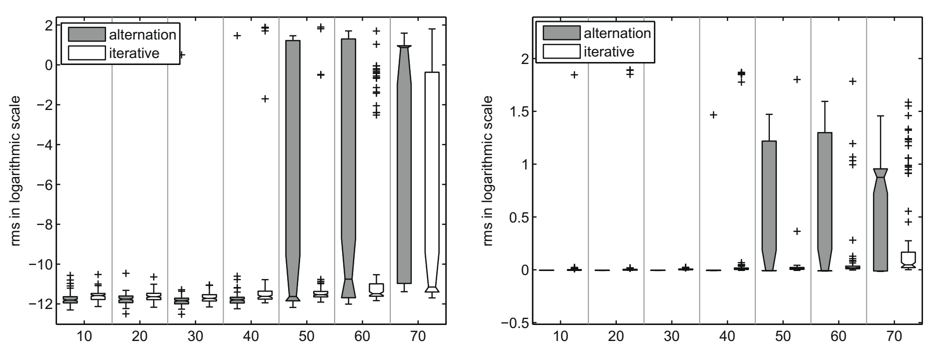

Fig. 7. Cylinder scene; *rms* in logarithmic scale, for different percentages of missing data: (left) no noise; (right) $\sigma = 1$.

(medians are represented by horizontal lines in thinner regions) and the crosses correspond to outliers. It should be highlighted that these boxplots are more representative than the mean value used in Ref. [17]. The reported experiments correspond to the no noise case (left) and to the case where a Gaussian noise of a *standard deviation* $(\sigma) = 1$ and zero mean (right) is added.

It can be seen that in general the Alternation applied to the matrix filled in with the iterative scheme performs better than applied directly to the originally given input W. When the percentage of missing data is below or equal to 40% (Fig. 7), no improvements are obtained by using the iterative scheme since the Alternation performs quite well with such amount of missing data. With more than 40% of missing data, the *rms* obtained when the Alternation is applied directly to the input matrix becomes higher than when it is applied to the matrix filled in with the iterative multiresolution scheme. Fig. 7(right) shows that the difference between the results obtained by applying Alternation to the matrix filled in by the iterative scheme and to the original input one is not as marked as in the free noise case (Fig. 7(left)).

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. 8 shows the obtained values. It can be seen that compared to *rms*, the rms_{all} is in general higher, which means that the initially known entries are better recovered than the missing ones in most cases. Notice that the difference between *rms* and rms_{all} is higher when the Alternation is applied directly to

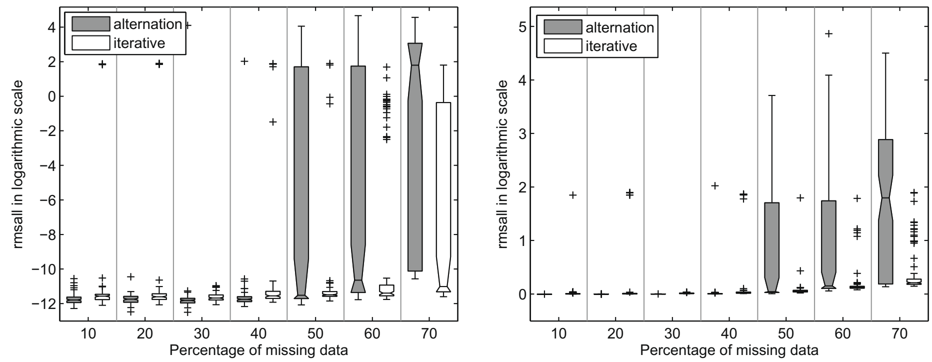

Fig. 8. Cylinder scene; rms_{all} in logarithmic scale, for different percentages of missing data: (left) no noise; (right) $\sigma = 1$.

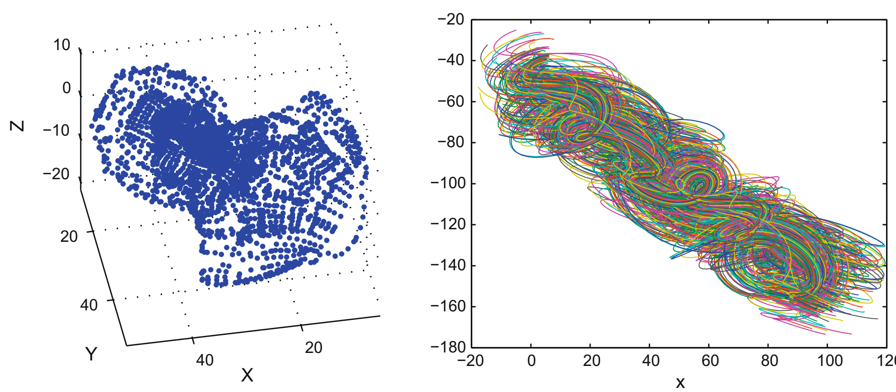

Fig. 9. Synthetic object: (left) Beethoven's sculpture; (right) feature point trajectories represented in the image plane.

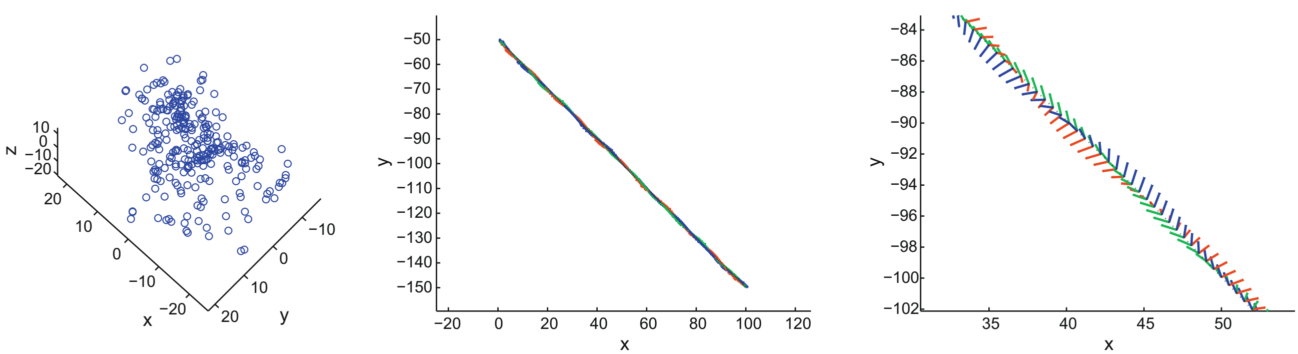

Fig. 10. Beethoven's sculpture scene: (left) 3D reconstruction; (middle) recovered camera motion; (right) zoom in the recovered camera motion.

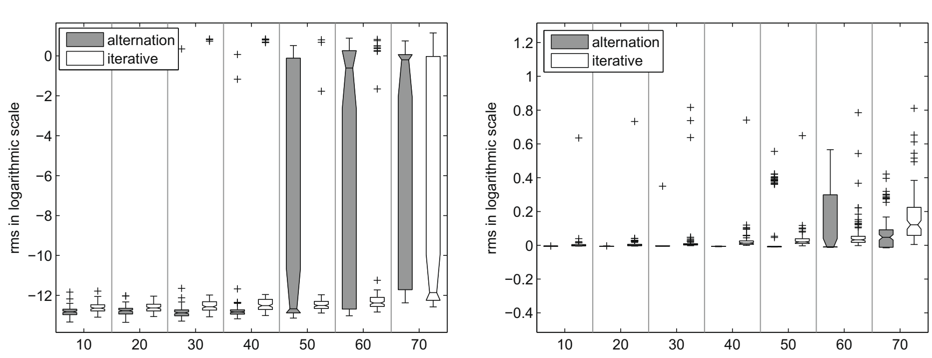

Fig. 11. Beethoven scene; *rms* in logarithmic scale, for different percentages of missing data: (left) no noise; (right) $std = 1$.

the original input matrix than when it is applied to the matrix filled in with the iterative scheme.

Experimental results in Ref. [17] show that the S and M factors are better recovered when the Alternation is applied to the matrix filled in with the iterative scheme than when the Alternation is applied to the original input matrix (see [17] for more details).

The second sequence containing a single object is generated by considering 200 frames and 266 trajectories from the second data set. Feature point full trajectories are plotted in Fig. 9(right). Fig. 9(left) contains a large amount of 3D points (about 2655) in order to visualize better the object. Fig. 10 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).

The obtained *rms* in this second sequence are plotted in Fig. 11. In the case of no noise, Fig. 11(left), it can be seen that for percentages of missing data higher than 40%, the Alternation gives in general smaller *rms* applied to the matrix filled in with the proposed iterative scheme, than to the original input one.

Working with noisy data, the *rms* is smaller when the Alternation is applied to the matrix filled in with the iterative scheme for percentages of missing data between 50% and 60% for $\sigma = 1$ (Fig. 11(right)).

Fig. 12 shows the obtained rms_{all}. As in the previous sequence, it can be seen that the rms_{all} is in general higher than *rms* and, again, the difference between *rms* and rms_{all} is higher when the Alternation is applied directly to the original input matrix. In particular, the rms_{all} is smaller when the Alternation is applied to the matrix filled in with the proposed iterative scheme, while the percentage of missing data is higher than 40% (Fig. 12(left) and (right)).

3.1.2. Multiple object case

For simplicity, a sequence containing only two objects is considered in the multiple object case. Sequences with more than two objects can be studied analogously. The main drawback of the multiple object case is that the rank of the matrix of trajectories W is not known and it must be estimated before applying a factorization technique. As mentioned above, the method proposed in Ref. [16] is used to estimate the rank of W, when the percentage of missing data is lower than 20%. Then, this estimated rank value is used through the whole experiments.

Once the matrix of trajectories has been filled in (partially or totally), trajectories belonging to the same object should be first clustered together. Then, the structure and motion corresponding to each object could be obtained by applying a single object SFM technique.

The studied sequence contains two cylinders. The rank estimation technique proposed in Ref. [16] applied to matrices with low percentage of missing data gives a rank value of $r = 8$, which makes sense since both cylinders define a full-rank motion and move independently. A total of 248 features (122 correspond to the first cylinder and 126 correspond to the second one) are tracked over 200 frames (Fig. 13(left)). Fig. 13(right) shows the trajectories plotted in the image plane.

The *rms* obtained in this multiple object sequence are shown in Fig. 14. Fig. 14(left) shows that the *rms* is in general smaller when the Alternation is applied to the matrix filled in with the iterative multiresolution scheme, for percentages of missing data between 40% and 60%. Results obtained in the case of noisy data are plotted in Fig. 14(right). It can be seen that results obtained when the Alternation is applied to the original input matrix are better than

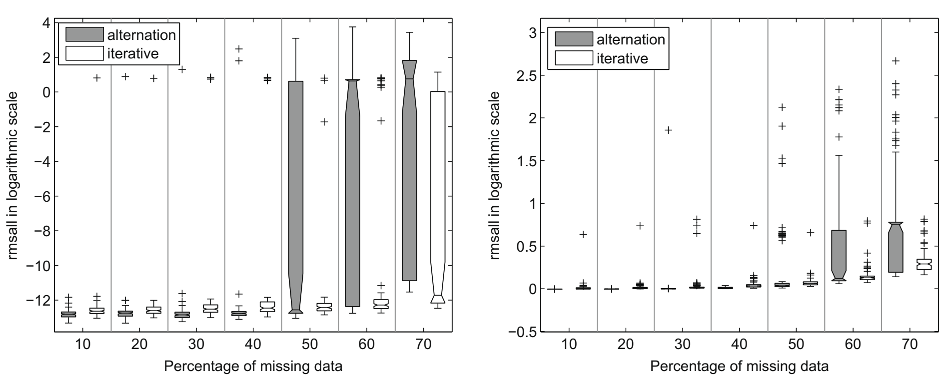

Fig. 12. Beethoven scene; rms_{all} in logarithmic scale, for different percentages of missing data: (left) no noise; (right) $\sigma = 1$.

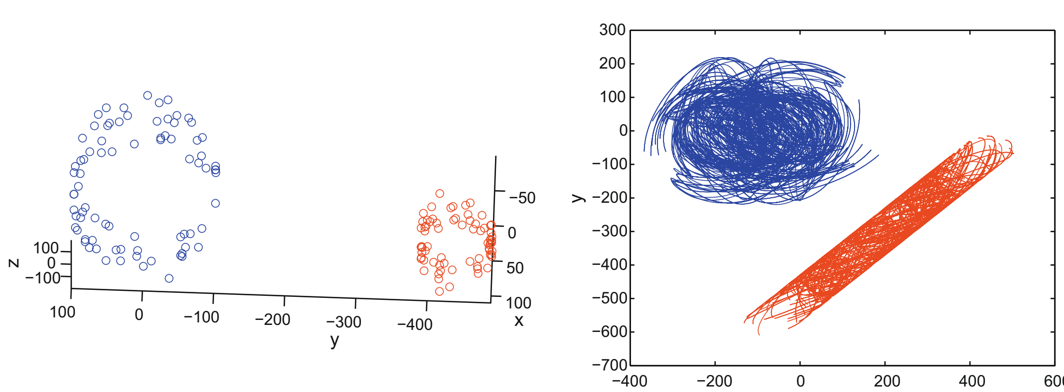

Fig. 13. Synthetic cylinders: (left) 3D position of the feature points in the first frame; (right) feature point trajectories plotted in the image plane.

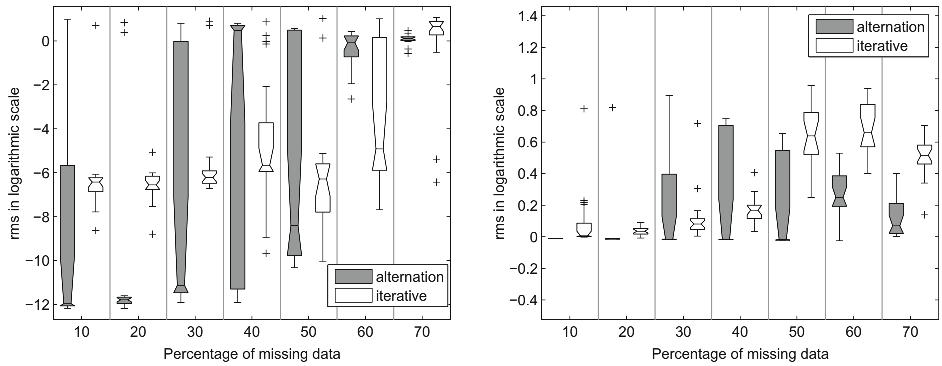

Fig. 14. Scene containing two cylinders: *rms* in logarithmic scale, for different percentages of missing data: (left) no noise; (right) $\sigma = 1$.

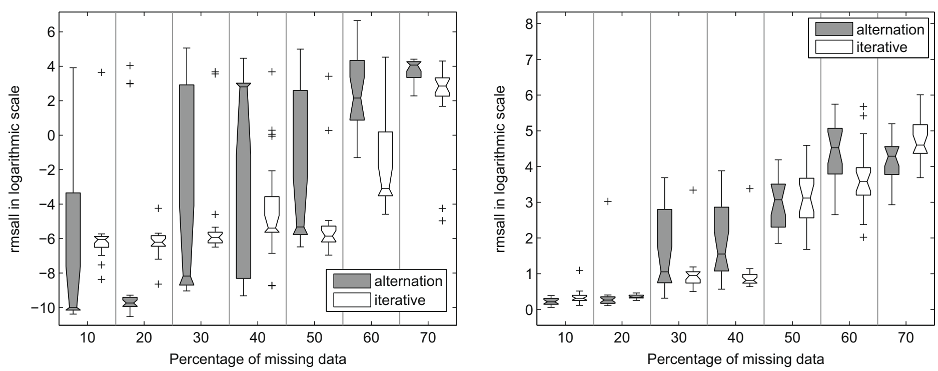

Fig. 15. Scene containing two cylinders: rms_{all} in logarithmic scale, for different percentages of missing data: (left) no noise; (right) $\sigma = 1$.

when applied to the matrix filled in with the proposed scheme, for any percentage of missing data.

Fig. 15(left) shows that the rms_{all} obtained when the Alternation is applied to the matrix filled in with the iterative multiresolution scheme are smaller than when it is directly applied to the original input matrix, while the percentage of missing data is higher than 30%. Notice that the *rms*, which only studies the initially known data, is always smaller when the Alternation is applied to the original input matrix (Fig. 14(right)) when working with noisy data. However, in this noisy data case, the rms_{all}, which studies the goodness of recovered missing entries, is smaller when the Alternation is applied to the matrix filled in with the proposed scheme, for percentages of missing data between 30% and 60%.

3.1.3. Summary

As a conclusion, it can be seen that in general, the Alternation applied to the original input matrix performs quite well, while the percentage of missing data is small. Therefore, it is not necessary to firstly apply the iterative multiresolution scheme in those cases. However, the results get 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. See, for instance, how the *rms* varies between 40% and 50% of missing data in Fig. 7(left). 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*.

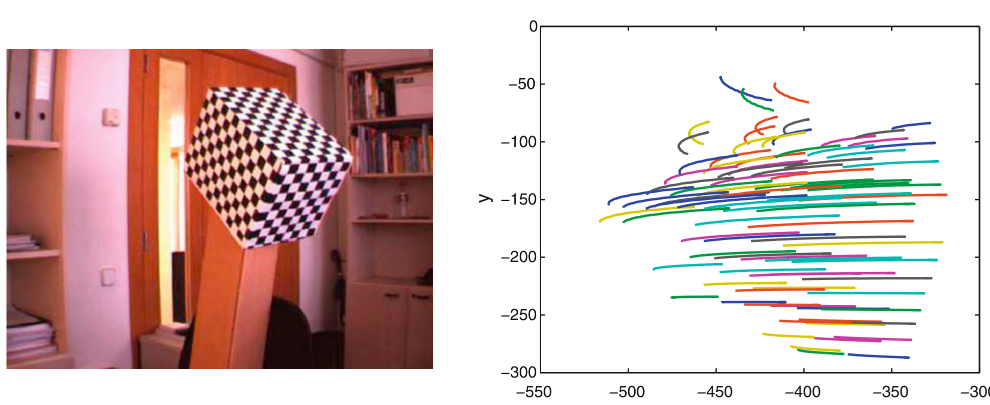

Fig. 16. (left) First object used for the real scene; (right) feature point trajectories represented in the image plane.

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.

3.2. 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 Figs. 16(left) and 19(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 Ref. [19]. As in the previous case, all the points are initially known in W_{all}, because only full trajectories are considered. The missing data are generated automatically by removing parts of random columns, as in the synthetic data experiments. 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 (Figs. 16(right) and 19(right)). Hence, the obtained matrices of trajectories are not of full rank and we have to deal with a degenerate case.

3.2.1. Single object case

The first sequence contains 87 points distributed over the squared-face-box shown in Fig. 16(left) tracked along 101 frames. Feature point trajectories are plotted in Fig. 16(right). Fig. 17 shows

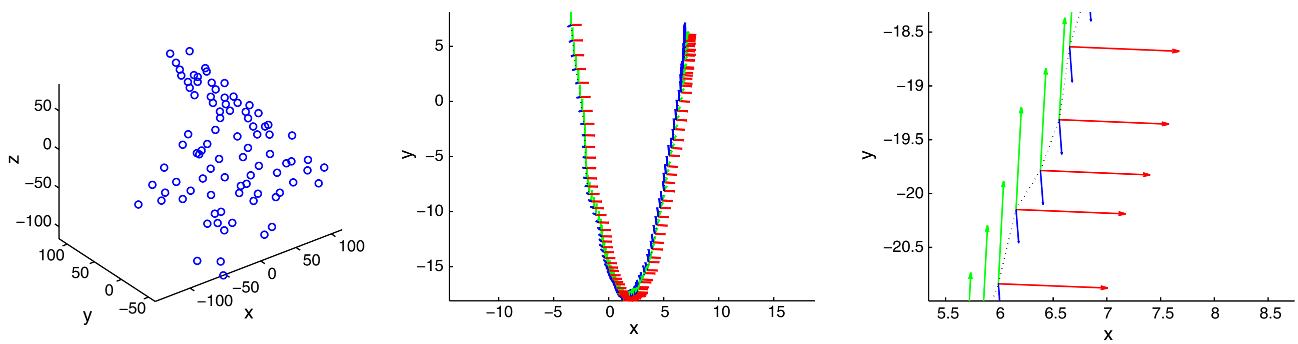

Fig. 17. First object; (left) 3D reconstruction; (middle) recovered camera motion; (right) zoom in the recovered camera motion.

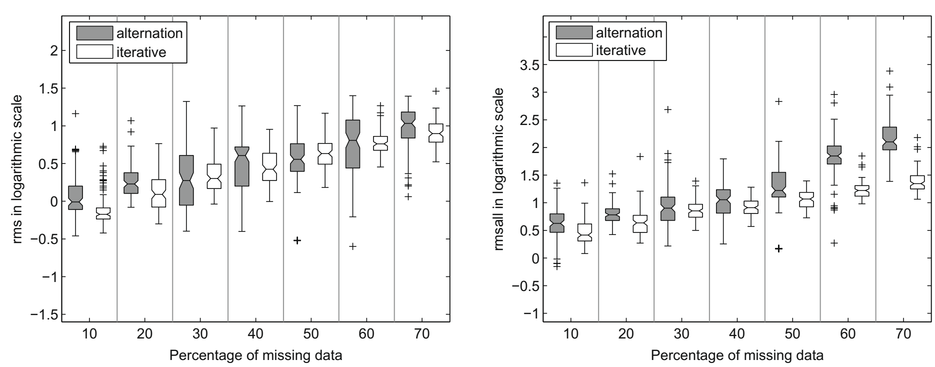

Fig. 18. First object: (left) *rms* in logarithmic scale, for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

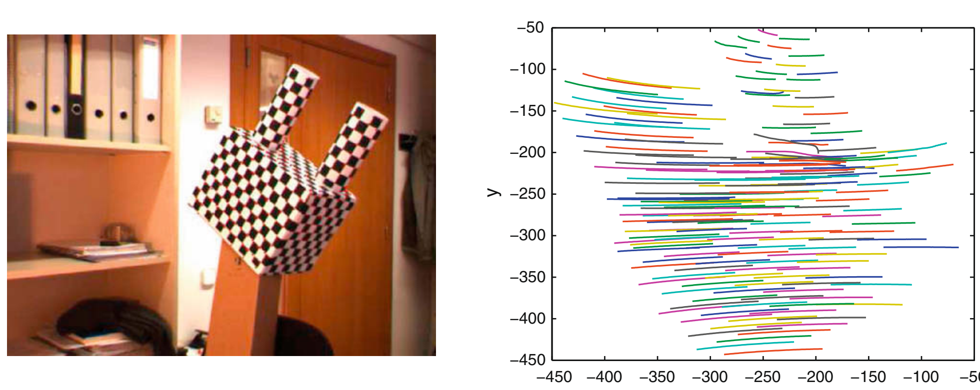

Fig. 19. (left) Second object used for the real scene; (right) feature point trajectories represented in the image plane.

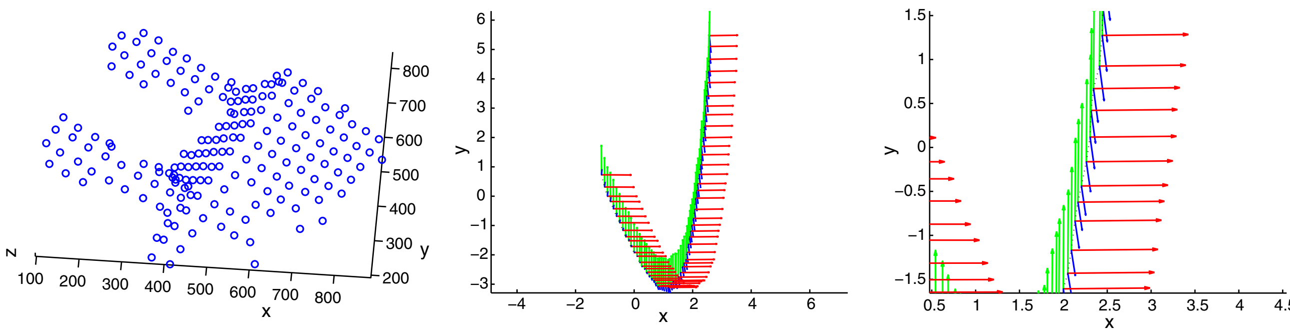

Fig. 20. Second object: (left) 3D reconstruction; (middle) recovered camera motion; (right) zoom in the recovered camera motion.

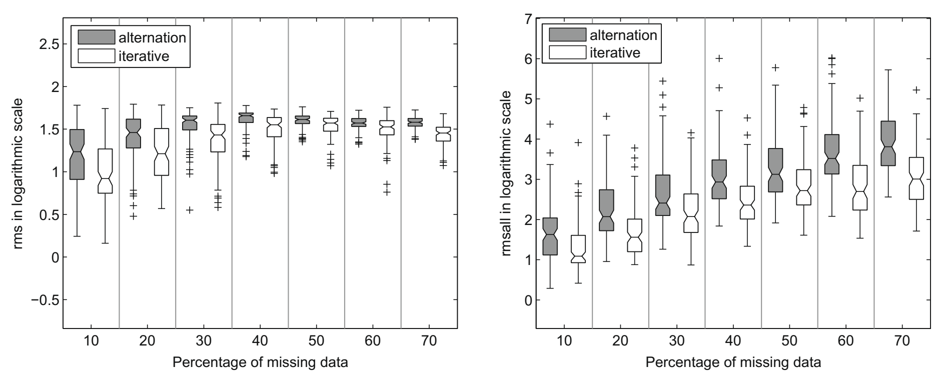

Fig. 21. Second object: (left) *rms* in logarithmic scale for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

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 about 10% of missing data.

The resulting *rms* obtained for different percentages of missing data are presented in Fig. 18(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. 18(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 distributed over the object shown in Fig. 19(left). Feature point trajectories are plotted in Fig. 19(right). Fig. 20 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. 21. 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.

3.2.2. Multiple object case

Full trajectory matrices corresponding to sequences of multiple objects are generated by merging different matrices of single object trajectories, after swapping x and y coordinates. Overlapping between objects are avoided for the sake of presentation simplicity by applying translations.

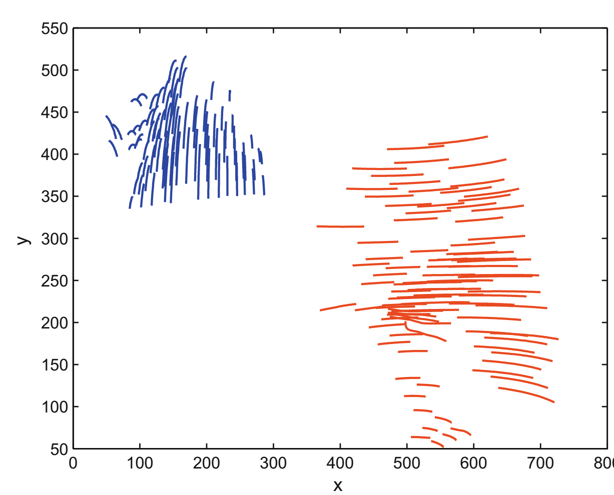

Fig. 22. Scene containing two objects: feature point trajectories plotted in the image plane.

As in the experiment with synthetic data, a sequence containing only two objects is presented. Sequences with more than two objects can be studied analogously. Concretely, a scene containing the objects shown in Figs. 16(left) and 19(left), is generated. It is defined by 61 frames and 181 feature points (87 from the first object and 94 from the second one). The obtained full feature point trajectories are plotted in the image plane in Fig. 22. The rank estimation technique proposed in [16] gives a rank value of $r = 5$, when the percentage of missing data is lower than 20%. Recall that both objects define a degenerate motion, as mentioned above.

Fig. 23 shows the obtained *rms* (left) and rms_{all} (right), respectively. It can be seen that the Alternation gives, in general, slightly

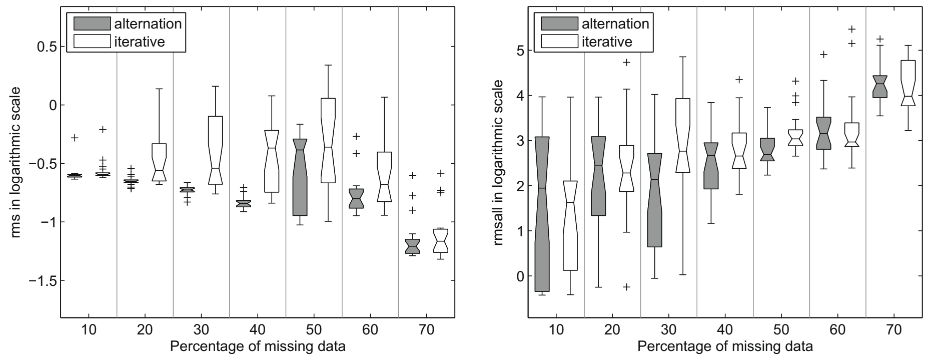

Fig. 23. Scene containing two objects: (left) *rms* in logarithmic scale for different percentages of missing data; (right) rms_{all} in logarithmic scale, for different percentages of missing data.

smaller *rms* values applied to the original input matrix than to one filled in with the proposed iterative scheme. On the contrary, the rms_{all} (see Fig. 23(right)) is in general smaller when the Alternation is applied to matrix filled in with the proposed scheme, instead of when it is directly applied to the original input matrix.

3.2.3. Summary

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, when the percentage of missing data is higher than 30%. Actually, in the single object case, it is better to apply the Alternation to the matrix filled in with the proposed scheme, even when the percentage of missing data is low.

4. Conclusions and future work

This paper presents an iterative multiresolution scheme for tackling the SFM problem with high percentages of missing data and for the single and multiple object cases. The idea of the iterative multiresolution scheme is to work with sub-matrices with a low percentage of missing data. Then, a factorization technique is applied to recover the missing entries with the product of the obtained factors. In the current work the Alternation technique is used to factorize these sub-matrices. 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 originally given input one, which has a lower percentage of known data. The main contribution of the current paper is the extension of the previously presented iterative multiresolution scheme to the multiple object case.

Experimental results considering synthetic and real data sequences that contain single and multiple objects are given. 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}.

As a future work, it would be interesting to use the iterative scheme with applications other than the SFM problem.

Acknowledgements

This work has been partially supported by the Spanish Government under Projects TRA2007-62526/AUT and DPI2007-66556-C03-03; research programme Consolider-Ingenio 2010: MIPRCV (CSD2007-00018); and Catalan Government under Project CTP 2008ITT 00001.

References

[1] H. Aanaes, R. Fisker, Robust factorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1215–1225.
[2] T.E. Boult, L.G. Brown, Factorization-based segmentation of motions, in: IEEE Workshop on Motion Understanding, 1991, pp. 179–186.
[3] A. Buchanan, A.W. Fitzgibbon, Damped Newton algorithms for matrix factorization with missing data, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, University of Oxford, 2005, pp. 316–322.
[4] P. Chen, 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 (2004) 1051–1063.
[5] J.P. Costeira, T. Kanade, A multibody factorization method for independently moving objects, IJCV: International Journal of Computer Vision (3) (1998) 159–179.
[6] S. Friedland, A. Niknejad, L. Chihara, A simultaneous reconstruction of missing data in DNA microarrays, Linear Algebra and its Applications 416 (2006) 8–28.
[7] G.H. Golub, C.F. Van Loan (Eds.), Matrix Computations, The Johns Hopkins University Press, 1989.
[8] R.F.C. Guerreiro, P.M.Q. Aguiar, Estimation of rank deficient matrices from partial observations: two-step iterative algorithms, in: EMMCVPR, 2003, pp. 450–466.
[9] N. Guilbert, A. Bartoli, A. Heyden, Affine approximation for direct batch recovery of euclidian structure and motion from sparse data, IJCV 69 (2006) 317–333.
[10] M. Han, T. Kanade, Reconstruction of a scene with multiple linearly moving objects, IJCV: International Journal of Computer Vision 53 (2000) 285–300.
[11] R. Hartley, F. Schaffalitzky, Powerfactorization: 3D reconstruction with missing or uncertain data, in: Australian-Japan Advanced Workshop on Computer Vision, 2003.
[12] D.W. Jacobs, Linear fitting with missing data for structure-from-motion, Computer Vision and Image Understanding, CVIU (1) (2001) 57–81.
[13] H. Jia, J. Fortuna, 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, 2006, pp. 1101–1007.
[14] C. Julià, A. Sappa, F. Lumbreras, J. Serrat, A. López, Factorization with missing and noisy data, in: Computational Science ICCS 2006: Sixth International Conference, LNCS, vol. 1, 2006, pp. 555–562.
[15] C. Julià, A. Sappa, F. Lumbreras, J. Serrat, A. López, An iterative multiresolution scheme for SFM, in: ICIAR 2006, LNCS 4141, vol. 1, 2006, pp. 804–815.
[16] C. Julià, A.D. Sappa, F. Lumbreras, J. Serrat, A. López, Rank estimation in 3d multibody motion segmentation, Electronics Letters 44 (2008) 279–280.
[17] 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 34 (3) (2009) 240–258.
[18] K. Kanatani, C. Matsunaga, Estimating the number of independent motions for multibody motion segmentation, in: ACCV: The Fifth Asian Conference on Computer Vision, 2002.
[19] Y. Ma, J. Soatto, J. Koseck, S. Sastry, An Invitation to 3D Vision: From Images to Geometric Models, Springer-Verlang, New York, 2004.
[20] D. Martinec, T. Pajdla, 3D reconstruction by fitting low-rank matrices with missing data, in: IEEE CVPR, vol. 1, 2005, pp. 198–205.
[21] T. Morita, T. Kanade, A sequential factorization method for recovering shape and motion from image streams, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (8) (1997) 858–867.

[22] D. Morris, T. Kanade, A unified factorization algorithm for points, line segments and planes with uncertainty models, in: International Conference on Computer Vision, 1998, pp. 696–702.
[23] T. Okatani, K. Deguchi, On the Wiberg algorithm for matrix factorization in the presence of missing components, International Journal of Computer Vision 72 (2007) 329–337.
[24] C.J. Poelman, T. Kanade, A paraperspective factorization method for shape and motion recovery, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 206–218.
[25] L. Quan, T. Kanade, A factorization method for affine structure from line correspondences, in: CVPR, 1996.
[26] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, International Journal of Computer Vision 9 (2) (1992) 137–154.
[27] B. Triggs, Linear projective reconstruction from matching tensors, Image and Vision Computing 15 (1997) 617–625.
[28] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman, Missing value estimation methods for DNA microarrays, Bioinformatics 17 (2001) 520–525.
[29] T. Wiberg, Computation of principal components when data is missing, in: Second Symposium of Computational Statistics, 1976, pp. 229–326.
[30] L. Zelnik-Manor, M. Irani, Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization, in: CVPR, vol. 2, 2003, pp. 287–293.