J Math Imaging Vis (2009) 34: 240–258
DOI 10.1007/s10851-009-0144-3

An Iterative Multiresolution Scheme for SFM with Missing Data

Carme Julià · Angel D. Sappa · Felipe Lumbreras · Joan Serrat · Antonio López

Published online: 21 February 2009

Abstract 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. Focussing on this problem, an iterative multiresolution scheme, which aims at recovering missing entries in the originally given input matrix, is proposed. Information recovered following a coarse-to-fine strategy is used for filling in the missing entries. The objective is to recover, as much as possible, missing data in the given matrix. Thus, when a factorization technique is applied to the partially or totally filled in matrix, instead of to the originally given input one, better results will be obtained. 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.

Keywords Factorization technique · Structure from motion

C. Julià (✉)
Intelligent Robotics and Computer Vision Group, Dept. of Computer Science and Mathematics, ETSE-URV, Av.Països Catalans 26, 43007, Tarragona, Spain
e-mail: carme.julia@urv.cat

A.D. Sappa
Computer Vision Center, UAB Edifici O Campus UAB, 08193 Bellaterra, Barcelona, Spain
e-mail: asappa@cvc.uab.es

F. Lumbreras · J. Serrat · A. López
Computer Vision Center, Computer Science Department, UAB Edifici O Campus UAB, 08193 Bellaterra, Barcelona, Spain

F. Lumbreras
e-mail: felipe@cvc.uab.es

J. Serrat
e-mail: joans@cvc.uab.es

A. López
e-mail: antonio@cvc.uab.es

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 [26] many variants have been presented in the literature (e.g. [24] for the case of paraperspective camera model; a sequential factorization method in [21]; [4] and [9] for the multiple object case, etc.). A brief review of factorization technique is provided below.

Let $\mathbf{p}_j$, with $j = 1, \dots, p$, be the 3D coordinates of feature points of a given object. At each frame $i = 1, \dots, f$, these feature points can be projected into the image plane by using an orthographic camera model:

$$\begin{aligned} u_{ij} &= \mathbf{i}_i^t \mathbf{p}_j + t_{xi}, \\ v_{ij} &= \mathbf{j}_i^t \mathbf{p}_j + t_{yi}, \end{aligned} \tag{1}$$

where $\mathbf{i}_i, \mathbf{j}_i$ correspond to the x and y camera axes at frame i and (t_{xi}, t_{yi}) its translation. Extensions considering line projections [25] and plane projections [22] have been also proposed in the literature. These 2D coordinates are stacked into the matrix of trajectories W, where every row represents a frame of the sequence and every column represents a given feature. By using (1), the following decomposition can be obtained:

$$W = \begin{bmatrix} u_{11} & \dots & u_{1p} \\ \vdots & & \vdots \\ u_{f1} & \dots & u_{fp} \\ v_{11} & \dots & v_{1p} \\ \vdots & & \vdots \\ v_{f1} & \dots & v_{fp} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^t & t_{x1} \\ \vdots & \vdots \\ \mathbf{i}_f^t & t_{xf} \\ \mathbf{j}_1^t & t_{y1} \\ \vdots & \vdots \\ \mathbf{j}_f^t & t_{yf} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 & \dots & \mathbf{p}_p \\ 1 & \dots & 1 \end{bmatrix} = M_{2f\times 4} S_{4\times p}, \quad (2)$$

where f and p are the numbers of frames and feature points, respectively.

Factorization techniques aim at expressing the matrix of trajectories W 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:

$$W_{2f\times p} = M_{2f\times r} S_{r\times p}, \quad (3)$$

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 [6]. The *Singular Value Decomposition* (SVD) gives the closed-form solution to this problem, when there are not missing entries, by using the fact that W is rank deficient. Hence, given the Singular Value Decomposition of W: $W = U_{2f\times r}\Sigma_{r\times r}V_{r\times p}^t$, motion and shape matrices can be obtained by:

$$M = U_{2f\times r}\Sigma^{\frac{1}{2}}, \qquad S = \Sigma^{\frac{1}{2}} V_{r\times p}^t. \quad (4)$$

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

$$\hat{M} = MA, \qquad \hat{S} = A^{-1}S \quad (5)$$

being M and S the true motion and shape matrices. The matrix A can be computed by imposing orthonormality between the two camera axes at each frame. That is, by solving the following $3f$ non-linear equations on the terms of A:

$$\begin{aligned} &\mathbf{i}_i^t AA^{-1}\mathbf{i}_i = 1, \\ &\mathbf{i}_i^t AA^{-1}\mathbf{j}_i = 0, \\ &\mathbf{j}_i^t AA^{-1}\mathbf{j}_i = 1, \quad i = 1, \dots, f. \end{aligned} \quad (6)$$

This final step is referred to as *normalization*. According to [26], this is a simple data fitting problem which, though non-linear, can be solved efficiently and reliably.

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. Since the SVD cannot be used with missing data, other methods have been proposed in the literature to tackle 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 [15] 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 initialization by other approaches. An iterative algorithm for recovering missing components in a large noisy low-rank matrix is provided by Chen and Suter [3]. 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 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 [16], Jia et al. present an algorithm that aims the SFM recovery with noisy and missing data. It is similar to the aforementioned one [15], 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 construct the final matrix.

Wiberg [28] presents an algorithm which 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. In a recent paper, Okatani et al. [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., [2, 7]), as the origin of what is called the *Alternation* technique. This iterative technique starts with an initial random factor S_0 or M_0 and, at each iteration k, computes alternatively each of the factors M_k and S_k, until the product M_kS_k converges to W. The key point of this 2-step algorithm is that, since the updates of S given M (analogously in the case of M given S) can be independently done for each row of M (or column of S), missing entries in W correspond to omitted equations. Due to that fact, with a large amount of missing data the method would fail to converge. Several variants of this approach have been proposed in the literature. In [7], 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 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, M and S factors correspond to the motion and shape 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 [1], Aanaes and Fisker present an Alternation-based scheme that can deal with mismatched features, missing features and noise on the features. Huynh et al. [13] present an outlier correction scheme, based on the Alternation technique, that iteratively updates the elements of the matrix of trajectories. Thus, the method corrects the outliers and factorizes the matrix of trajectories simultaneously. In [2], 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 known data are used. The formulation is similar to the one presented in [15]. The main difference is that the problem is formulated in terms of the original subspaces, while in [15] the complementary ones are used. Finally, Guilbert et al. [8] present a batch method for recovering 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.

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 introduced in [17]. Improvements to this approach were presented in [18]. The key point of the implemented 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, for each set (sub-matrix), the 3D reconstruction and the camera motion corresponding to the used feature points are obtained by applying a factorization technique. The missing entries in each selected set can be filled in just by multiplying the recovered shape and motion matrices. One of the main contributions of this paper over the two preliminary approaches ([17] and [18]) is that only sub-matrices with a percentage of missing data below 50% are used, in order to assure good recovered factors. Furthermore, a more extensive 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, by using a robust RANSAC [5] based strategy. 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 matrix of trajectories is initially known.

The proposed approach should be seen as a pre-processing technique; that is, firstly the originally given input 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 previously filled in with the proposed scheme, instead of applying it directly to the originally given input matrix, which contains a higher percentage of missing data.

Another constraint of most of the current factorization techniques is that the matrix of trajectories is assumed to be outlier-free, as pointed out in [14] and [13]. This is not always a realistic assumption, since outliers can appear in real sequences due to failures in the tracker. In [14] and [13], they propose approaches to detect and correct outliers, respectively. As mentioned above, the current paper is focused on dealing with high percentages of missing data in the matrix of trajectories; the problem of detection and correction of outliers is out of scope of this work.

The remainder of the paper is organized as follows. Section 2 introduces the iterative multiresolution scheme. Section 3 provides an evaluation study of the performance of the proposed scheme, both for synthetic and real data. Three different factorization techniques are considered in the study: the *Alternation with motion constraints*, the Powerfactorization and the Alternation/Damped Newton Hybrid. Conclusions and future work are given in Sect. 4.

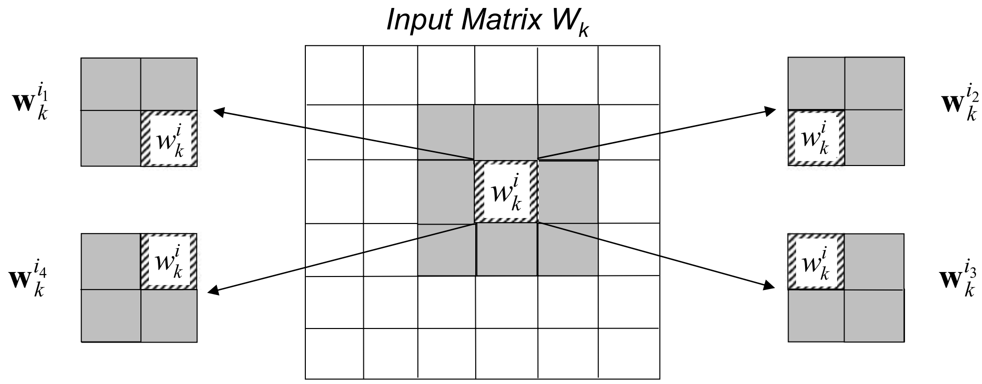


Fig. 1 w_k^i and its four corresponding $\mathbf{w}_k^{i_n}$ matrices, computed during the first stage, at iteration $k = 6$

2 Proposed Approach: Iterative Multiresolution Scheme

Essentially, the basic idea of the proposed approach is to work with sub-matrices that contain 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 sub-matrices and the missing data are filled in with the resulting product MS. The proposed approach consists of two stages, which are described below.

2.1 Observation Matrix Splitting

Let $W_{2f\times p}$ be the matrix of trajectories (also referred to through the paper as originally given 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 sub-matrices with a reduced percentage of missing data. This splitting process consists of the following two steps:

– *Splitting:* in the first step, the given input matrix W is split into a set of $k \times k$ non-overlapped sub-matrices,[1] as can be seen in Fig. 1 in the case of $k = 6$. Each obtained sub-matrix is defined as w_k^i, being $i = 1, \ldots, k^2$, and has a size of $\lfloor \frac{2f}{k} \rfloor \times \lfloor \frac{p}{k} \rfloor$ (e.g., see the sub-matrix w_k^i in Fig. 1). For the sake of presentation simplicity, hereinafter the split 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. Furthermore, if some of the $k \times k$ sub-matrices are discarded due to a high percentage of missing data, a poor partition of W could be obtained.

[1]Notice that the algorithm begins with $k = 2$; otherwise no partition is obtained.

Hence, in order to both improve the confidence of recovered data and also obtain a richer partition of W, in this second step a multiresolution approach is followed (only when $k > 2$). This multiresolution consists in computing four overlapped sub-matrices $\mathbf{w}_k^{i_n}$, $n = 1, \ldots, 4$, with twice the size of w_k^i (see Fig. 1) for every w_k^i.

The idea of this enlargement process is to study the recovered entries in w_k^i when different size regions are considered. Hence, entries are not recovered from a single small sub-matrix and an overlapping among filled in entries is provided. 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. Note that most of current sub-matrix factorization based approaches follow fine-to-coarse strategies relying on an initial full sub-matrix; then results are used in a kind of region growing scheme.

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 into four different ways, by shifting $\mathbf{w}_k^{i_n}$ half of its size through rows, columns or both at the same time. Figure 2 illustrates the five partitions of matrix W_k generated at the sixth iteration—i.e., the one generated by all the w_k^i and the remainder four ones, obtained with all the $\mathbf{w}_k^{i_n}$ sub-matrices. When all these matrices are considered together, the overlap between different areas is obtained, see textured cell in Fig. 1 and Fig. 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 technique at every single sub-matrix.

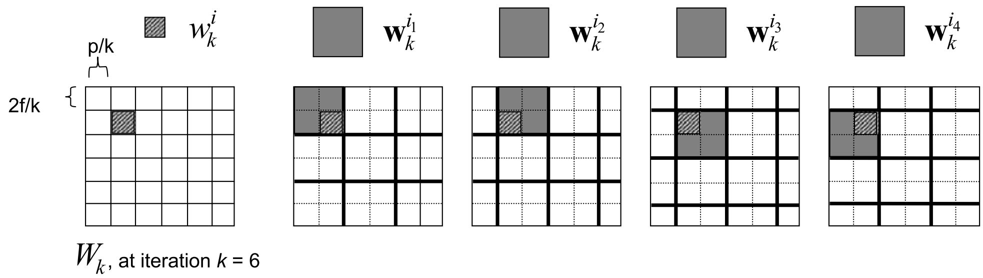


Fig. 2 Five partitions of matrix W_k. Note the overlap between a w_k^i sub-matrix with its corresponding four $\mathbf{w}_k^{i_n}$ sub-matrices, computed during the first stage

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 less than 50% of missing data are considered).

Independently of their size hereinafter sub-matrices will be referred to as W_s. Therefore, given a sub-matrix W_s its corresponding M_s and S_s matrices are obtained by using a factorization technique. Then, the product $M_s S_s$ is used to fill in the corresponding matrix W_s. Finally, the root mean squared (*rms*) is computed as follows:

$$rms_s = \frac{\|W_s - M_s S_s\|_F}{\sqrt{n}} = \sqrt{\frac{\sum_{i,j}|(W_s)_{ij} - (M_s S_s)_{ij}|^2}{n}}, \quad (7)$$

where i and j correspond to the index pairs of the known entries in $(W_s)_{ij}$ 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 $1/rms_s$.

In our original approach [17] and [18], a threshold ς was defined and used to discard the recovered entries in W_s when its corresponding rms_s was higher than ς. However, the main drawback was to find the best value for that threshold. One of the main improvements of the current paper over the original approach 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 sub-matrix W_s has been processed, recovered missing data are used for filling in the originally given input matrix W. There are two kind of entries in this merging step: the initially missing ones and the initially known ones. The first ones, can be recovered from more than a single sub-matrix. In this case, each missing entry is filled in with the normalized weighted average of the corresponding recovered values. Concretely, the aforementioned $1/rms_s$ is used as a weight to measure the goodness of the recovered entry. If a missing entry is recovered from only one sub-matrix, the filled in entry takes directly the corresponding value. The second kind of entries, the initially known ones, could be modified in the merging step; the mean between the original entry and the weighted average of the corresponding recovered values from each sub-matrix gives the final value of the entry.

Once recovered missing data have been used for filling in the input matrix W, the iterative process starts again (Sect. 2.1) splitting the new matrix W 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 totally filled. Figure 3 presents a chart flow illustrating the stages of the algorithm; an outline of it is given below:

Outline of the algorithm

Inputs: W trajectory input matrix; *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 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 $\mathbf{w}_k^{i_n}$, $n = 1, \dots, 4$).
3. Sub-matrices processing: apply a factorization technique to all the sub-matrices.
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 Evaluation Study

The aim at this stage is to study the robustness to missing and noisy data of a factorization technique applied to the

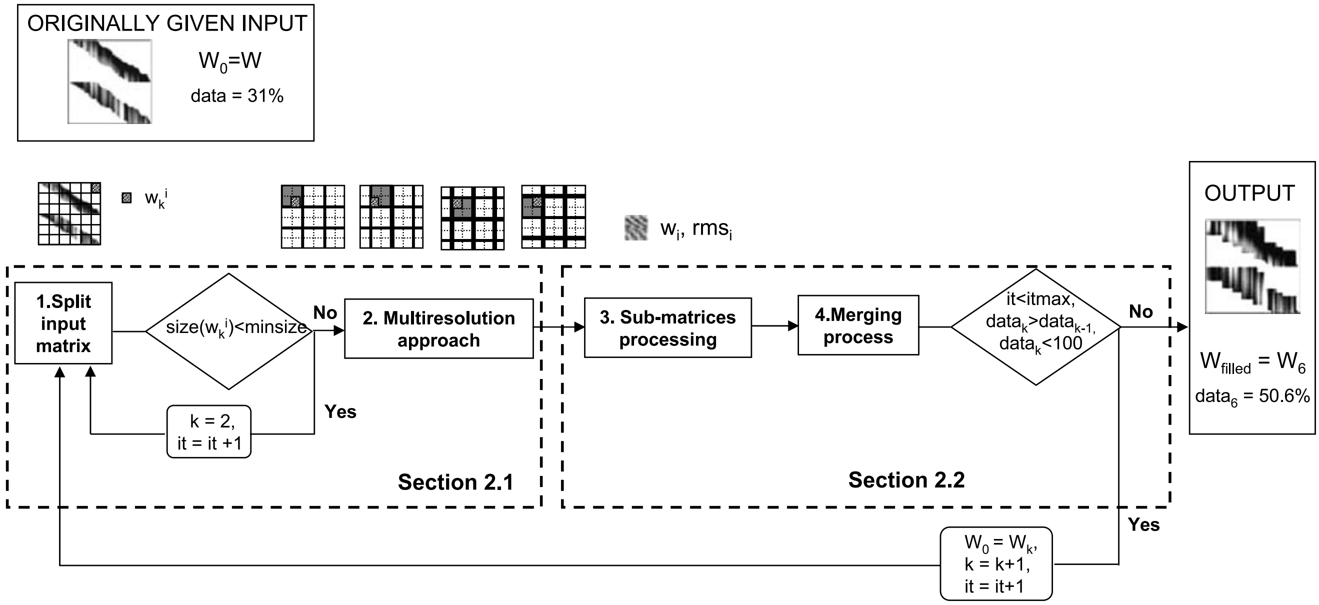


Fig. 3 Algorithm. Example of an originally given input matrix W with only 31% of known data. The obtained filled matrix W_{filled} after 6 iterations contains 50.6% of known data

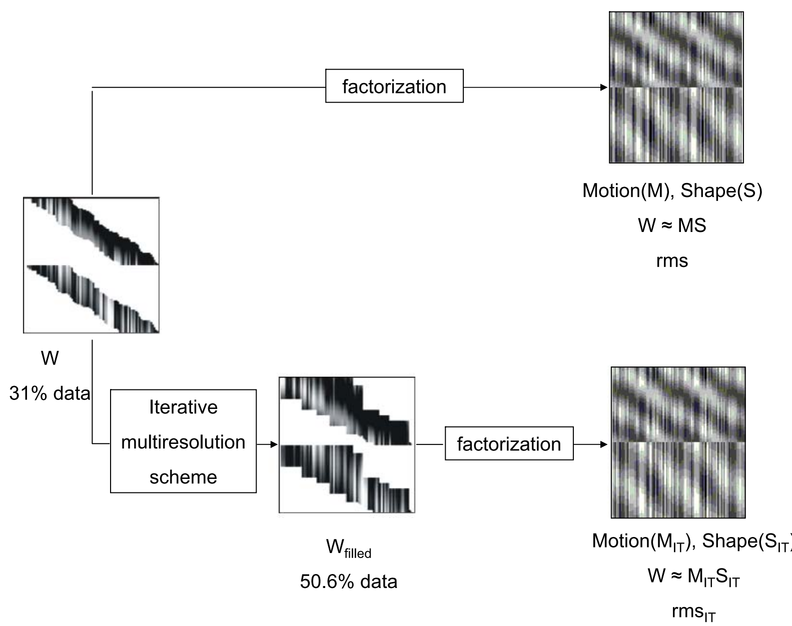


Fig. 4 Evaluation study: the recovered factors and the obtained *rms* in each case are compared. IT stands for iterative scheme. In this example, the originally given input matrix only contains 31% of known data

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 originally given 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 originally given input matrix W;
- a factorization technique over the matrix filled in with the proposed iterative multiresolution scheme W_{filled}. Hence, the considered factorization 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.

Experiments using both synthetic and real data are presented below. The following factorization techniques are used in the evaluation study: a) *Alternation with motion constraints*, which is presented in Sect. 3.2; b) Powerfactorization [11]; and c) Alternation/Damped Newton Hybrid [2]. Actually, experimental results are focused on the first technique (see Sect. 3.2.1 and Sect. 3.2.2). The other two methods are included in order to show that the proposed scheme is also valid with other factorization techniques. Results obtained in these cases are depicted in Sect. 3.3. Matrices of trajectories used during the experiments are assumed to be outlier-free, since none of the techniques used to evaluate the proposed scheme are robust to outliers (as mentioned in [14] and [2]).

3.1 Error Computation

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 originally given 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:

$$rms = \frac{\|\widehat{W} - MS\|_F}{\sqrt{n}} = \sqrt{\frac{\sum_{i,j} |(\widehat{W})_{ij} - (MS)_{ij}|^2}{n}}, \quad (8)$$

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. This strategy is used in both synthetic and real data experiments. 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 depends on 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 appropriated for an evaluation study.

As pointed out in [3], the *rms* defined by the expression (8) 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 originally given input W, but it ignores how missing entries are recovered. Since all entries are initially known in W_{all}, the proposed evaluation study is performed by computing the root mean square error considering all the entries in W_{all}. Hereinafter, this measure will be referred to as rms_{all} and it is defined as follows:

$$rms_{all} = \frac{\|W_{all} - MS\|_F}{\sqrt{2fp}}, \quad (9)$$

where $2fp$ is the number of elements in the matrix W_{all}.

Furthermore, and only in the synthetic case, the recovered M and S are compared to the ground truth matrices (M_G and S_G) with the following strategy. First of all, in order to compute the error between the recovered shape matrix S and the ground truth S_G, they have to be represented in the same reference system. In other words, a 4×4 transformation matrix H, such that $S_G = HS$, should be computed. Since the correspondence between columns in S and S_G is known, matrix H could be directly computed by using any quartet of four columns. However, elements in S (i.e., columns) may correspond to wrongly recovered data. Therefore, a robust RANSAC [5] based strategy is proposed to compute the transformation matrix H as indicated below.

Random sampling. Repeat the following three steps N times (in our experiments N was set to 10).

1. Select a random sample μ of 4 different columns of S.
2. For this subsample μ, compute H_μ using the pseudoinverse [12] of S_μ, which gives the shortest length least squares solution to the problem $S_G = H_\mu S_\mu$. Therefore, $H_\mu = S_G S_\mu^t (S_\mu S_\mu^t)^{-1}$.
3. For this solution H_μ, compute both $H_\mu S$ and the number of inliers among the entire set of columns of S. An inlier is defined by a column j, where $\|S_{Gj} - H_\mu S_j\|_F < \sigma$, where $\sigma = \frac{\|S_G\|_F}{\sqrt{4p}}$, being p the number of columns of S_G.

Solution:

1. Choose the solution that has the largest number of inliers. Let H_i be this solution.
2. Refine H_i by using its corresponding inliers, instead of only 4 points. Let H be this final refined transformation.

The transformation matrix H computed above with the robust technique is now used to compute the error of the estimated shape, using all the points of $S_{4\times p}$:

$$rms_S = \frac{\|S_G - HS\|_F}{\sqrt{4p}}. \quad (10)$$

The inverse of this transformation matrix is used to measure the error of the estimated motion $M_{2f\times 4}$:

$$rms_M = \frac{\|M_G - MH^{-1}\|_F}{\sqrt{2f4}}. \quad (11)$$

3.2 Iterative Multiresolution Scheme by Using Alternation with Motion Constraints

Due to its simplicity and its good performance, Alternation technique was used in [18], where it was compared to the Damped Newton technique [2]. It was shown that the first is more appropriated, both from the results and from the computational cost.

Focusing on the SFM problem, the *Alternation with motion constraints* is introduced in this section. It is a variant of the Alternation similar to the one mentioned above for the special case of SFM [11]. 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, 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 proposed 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:

$$M_k = W S_{k-1}^t (S_{k-1} S_{k-1}^t)^{-1}. \tag{12}$$

   Define $M = [\, R\ \mathbf{t}\,]$, 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; in other words, 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 [26]:

$$\begin{aligned} \mathbf{i}_i^t A A^{-1} \mathbf{i}_i &= 1, \\ \mathbf{j}_i^t A A^{-1} \mathbf{j}_i &= 1, \\ \mathbf{i}_i^t A A^{-1} \mathbf{j}_i &= 0, \end{aligned} \tag{13}$$

   where $\mathbf{i}_i$ and $\mathbf{j}_i$ are the x and y camera axes at frame i (see (2)), and $i = 1, \ldots, f$, being f the number of frames.

   Actualize M_k:

$$\widetilde{M}_k = M_k A. \tag{14}$$

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

$$\widetilde{\mathbf{w}}^j = \mathbf{w}^j - \widetilde{\mathbf{t}}_k, \quad j = 1, \ldots, p. \tag{15}$$

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

$$(S_R)_k = ((R)_k^t (R)_k)^{-1} ((R)_k^t \widetilde{W}). \tag{16}$$

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.

One of the main advantages of this two-step algorithm is that the updates of M given S (analogously S given M) can be done by solving a least squares problem for each row of M independently. Therefore, missing entries in W correspond to omitted equations and due to that fact obtained result depends on the number of initially known data in each row and column of W.

Results obtained, when the Alternation with motion constraints is considered in the proposed scheme, are presented in the next two sections, for synthetic and real data experiments, respectively.

3.2.1 Synthetic Data

This section provides results obtained with two data sets from different objects. The first data set is generated by randomly distributing 3D 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. 11 (left). Two different sequences are obtained with these objects by performing a rotation and a translation over each one 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 feature points. The trajectories are plotted in Fig. 5 (right). Figure 6 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 originally given input matrix contains about 20% of missing data, which have been removed from the initially known trajectory matrix W_{all}.

The obtained *rms* considering different percentages of missing data is shown in Fig. 7. Concretely, the mean of the computed *rms* is plotted, in logarithmic scale. The reported experiments correspond to the no noise case (left) and to cases with added Gaussian noise of a *standard deviation* (σ) of 1/3 (middle) and 1 (right), respectively.

It can be seen that in general the Alternation applied to the matrix previously filled in with the proposed iterative scheme (denoted as It-Alt in the plots) performs better than applied directly to the originally given input W (denoted as Alt). When the percentage of missing data is below or equal to 20% (Fig. 7 (left) and (middle)) or 30% (Fig. 7 (right)),

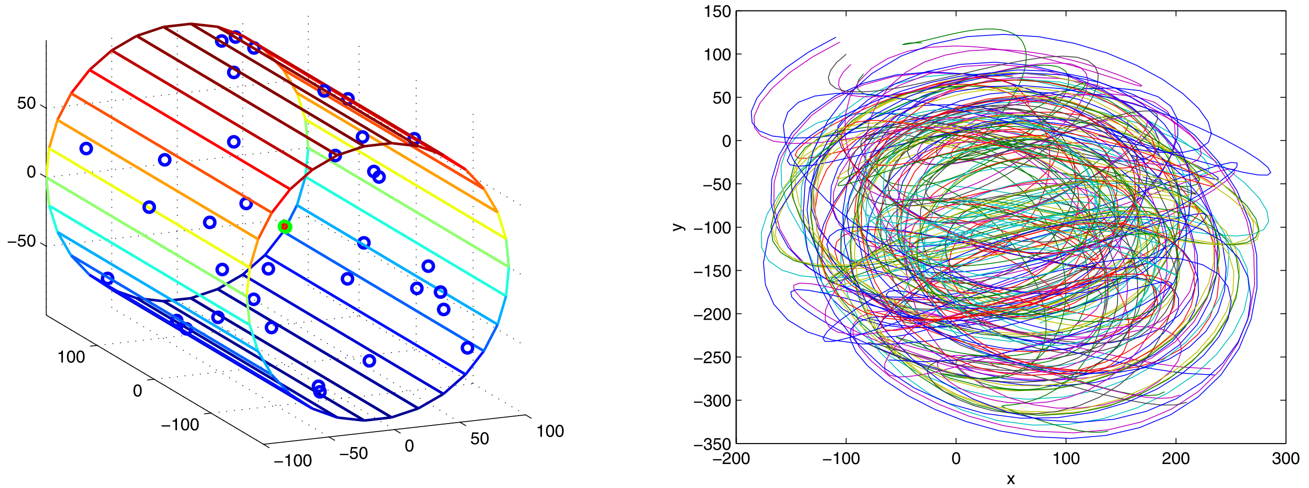


Fig. 5 Synthetic object: (*left*) cylinder; (*right*) feature point trajectories represented in the image plane

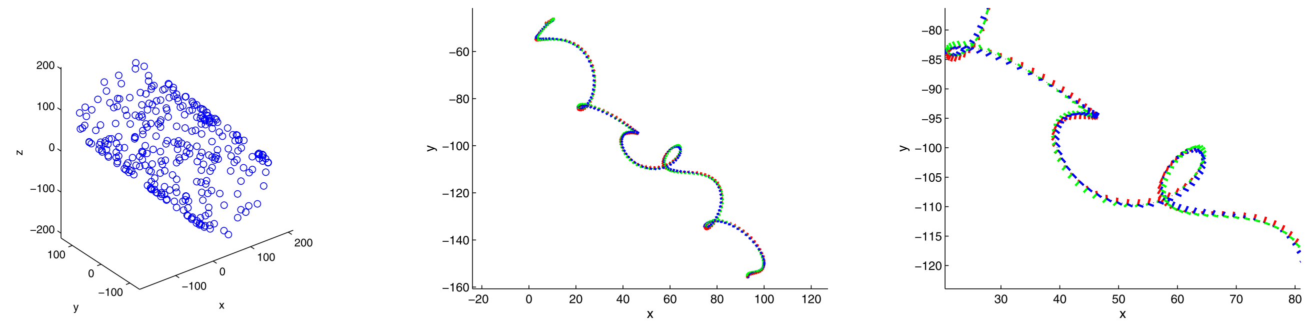


Fig. 6 Cylinder scene: (*left*) 3D reconstruction (thicker points correspond to reappearing features); (*middle*) recovered camera motion; (*right*) enlargement of the recovered camera motion

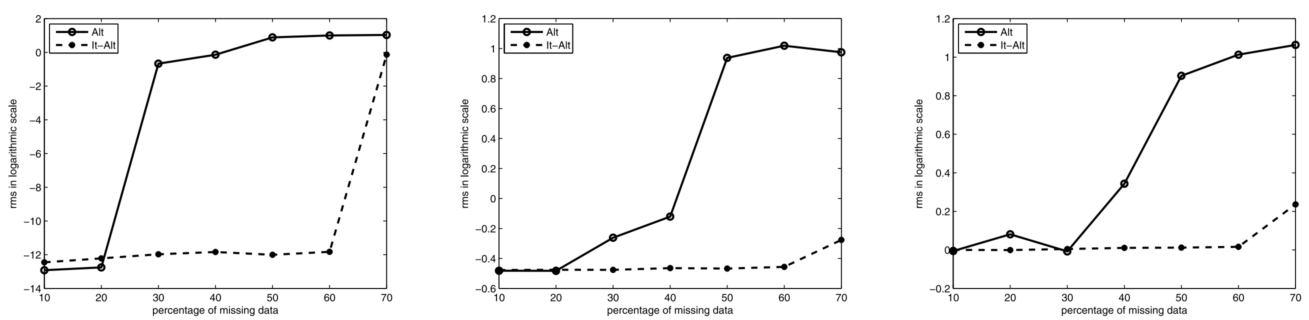


Fig. 7 Cylinder scene; mean of *rms* in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

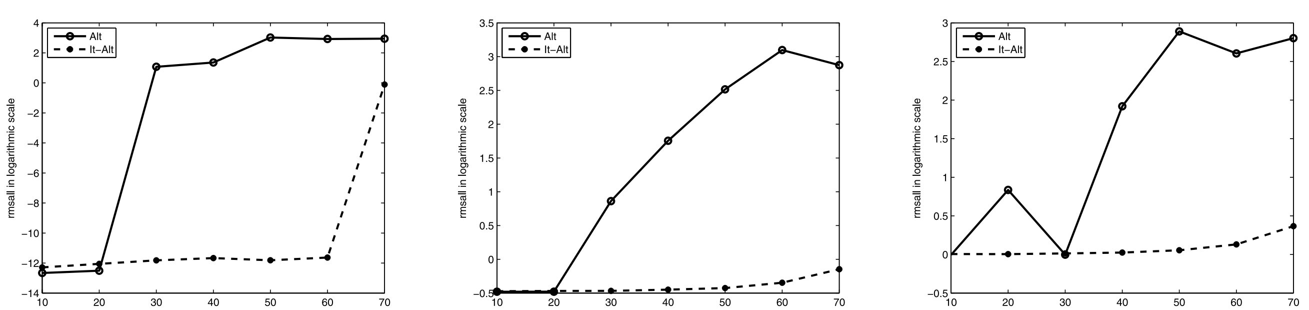


Fig. 8 Cylinder scene; mean of the rms_{all} in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

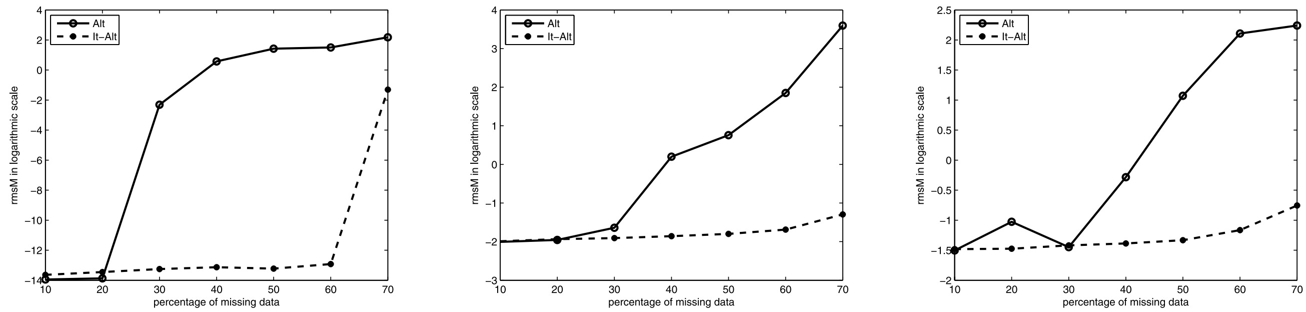


Fig. 9 Cylinder scene; mean of the error of the recovered motion (rms_M) in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

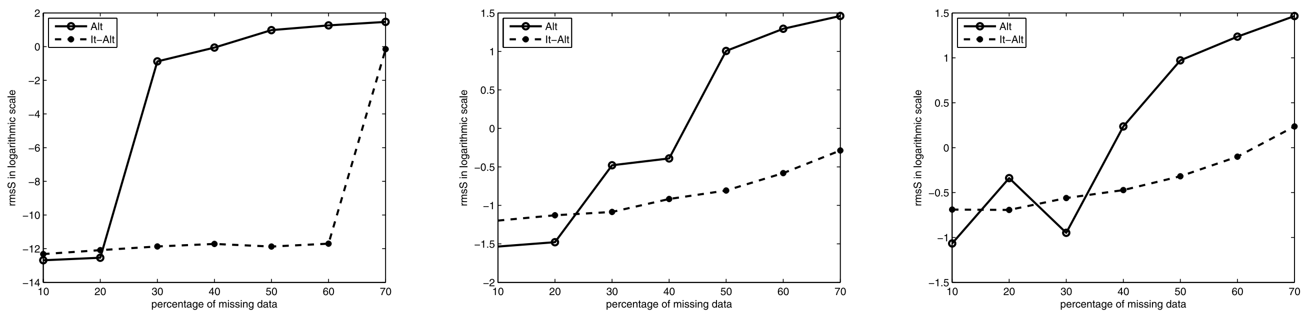


Fig. 10 Cylinder scene; mean of the error of the recovered shape (rms_S) in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

no improvements are obtained by using the iterative scheme since the Alternation performs quite well with such reduced amount of missing data.

On the contrary, with more than 20% (no noise and $\sigma = 1/3$ cases) and 30% ($\sigma = 1$) of missing data, the *rms* obtained when the Alternation is applied directly to the originally given input matrix becomes higher than when it is applied to the matrix filled in with the proposed iterative multiresolution scheme. In Fig. 7 (middle) and (right), it can be seen that the difference between the two approaches 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 originally given input matrix, instead of only the initially known ones, is also studied. Figure 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 the originally given input matrix than when it is applied to the matrix filled in with the iterative scheme.

The recovered factors S and M are also studied. The errors are measured by computing the rms_S (see (10)) and rms_M (see (11)), which are plotted in Fig. 9 and Fig. 10, respectively. It can be seen that the results are similar to the ones obtained with the *rms*: better results are obtained when Alternation is applied to the matrix previously filled in with the proposed iterative scheme, except in cases in which the percentage of missing data is below or equal to 20% (no noise and $\sigma = 1/3$ cases) or 30% ($\sigma = 1$ case).

In the second sequence, the numbers of frames and trajectories are 200 and 266, respectively. Feature point full trajectories are plotted in Fig. 11 (right). Figure 11 (left) contains a large amount of 3D points (about 2655) in order to visualize better the object. Figure 12 shows an example of the recovered shape (left) and motion ((middle) and (right)) obtained by applying Alternation to the matrix previously filled in with the proposed iterative scheme (again, originally given input matrix contains 20% of missing data).

The obtained *rms* in this second sequence is plotted in Fig. 13. In the case of no noise, Fig. 13 (left), it can be seen that for percentages of missing data between 10% and 70%, the Alternation gives smaller *rms* applied to the matrix filled in with the proposed iterative scheme, than to the originally given input one.

When noisy data are considered (Fig. 13 (middle) and (right)), the advantages of previously using the proposed scheme can be appreciated, while the amount of missing data is higher than 30% for $\sigma = 1/3$ (Fig. 13 (middle)) and for percentages between 30% and 70% for $\sigma = 1$ (Fig. 13 (right)).

Figure 14 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}

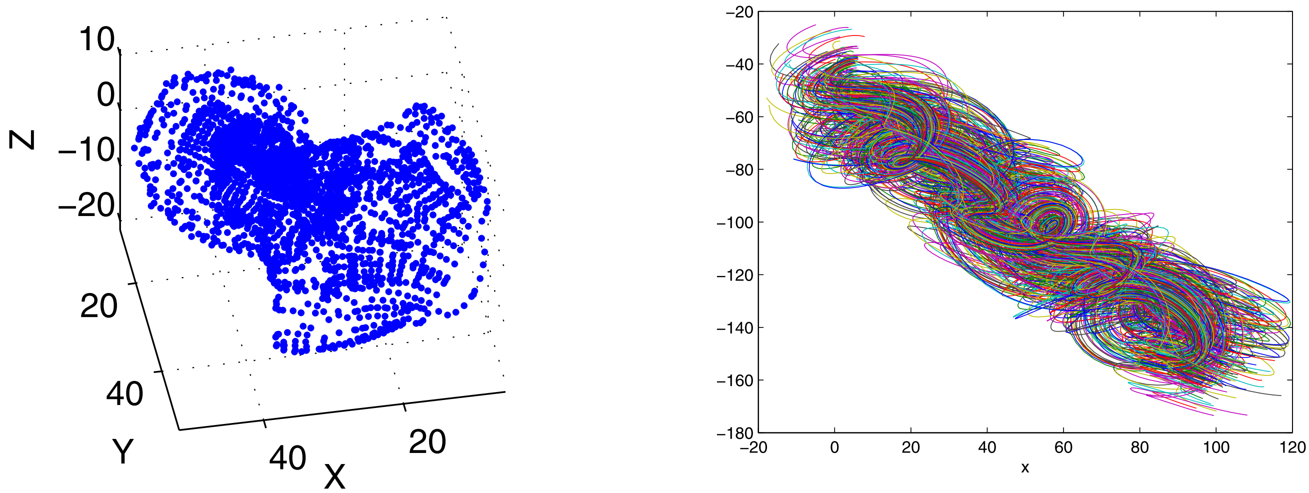


Fig. 11 Synthetic object: (*left*) Beethoven's sculpture; (*right*) feature point trajectories represented in the image plane

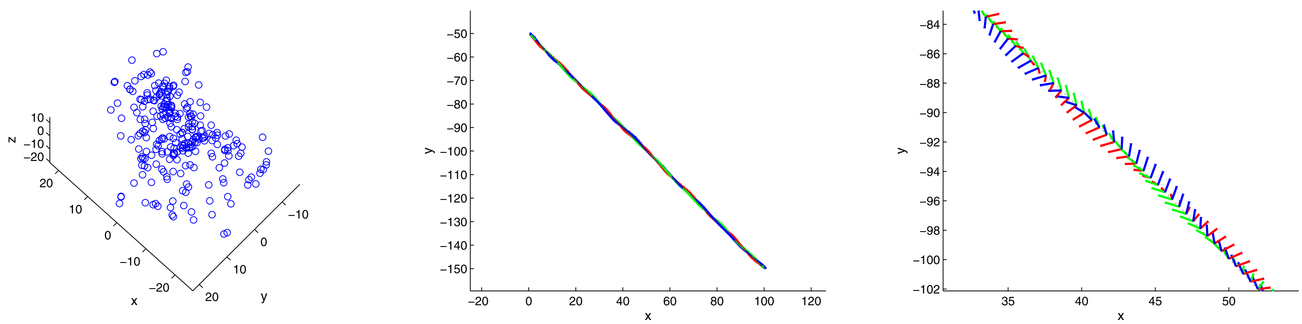


Fig. 12 Beethoven's sculpture scene: (*left*) 3D reconstruction; (*middle*) recovered camera motion; (*right*) enlargement of the recovered camera motion

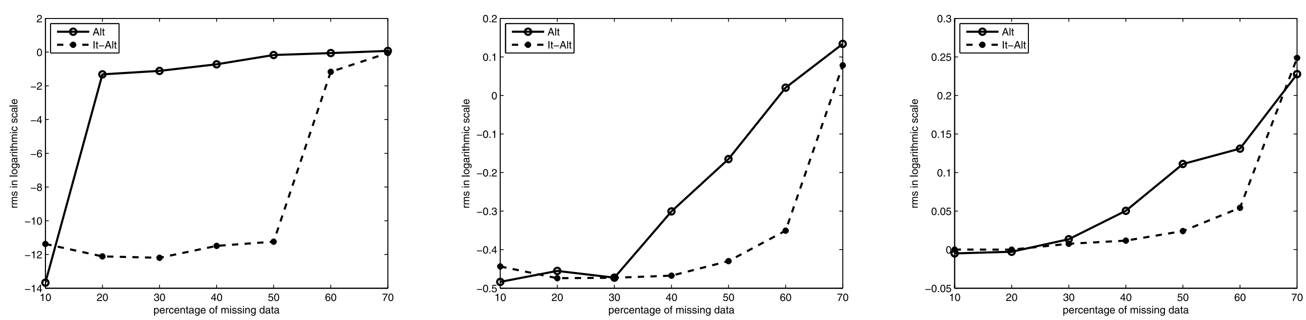


Fig. 13 Beethoven scene; mean of the *rms* in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

is higher when the Alternation is applied directly to the originally given 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 10% (Fig. 14 (left) and (middle)) and 30% (Fig. 14 (right)).

The rms_M is shown in Fig. 15. The results are very similar to the ones obtained in the *rms* study; the rms_M is smaller when the Alternation is applied to the matrix previously filled in with the proposed iterative scheme, while the percentage of missing data is higher than 10% (Fig. 15 (left) and (middle)) and 20% (Fig. 15 (right)).

In Fig. 16, it can be seen that the obtained rms_S is smaller when the Alternation is applied to the matrix previously filled in with the proposed iterative scheme, while the percentage of missing data is higher than 10% for the no noisy

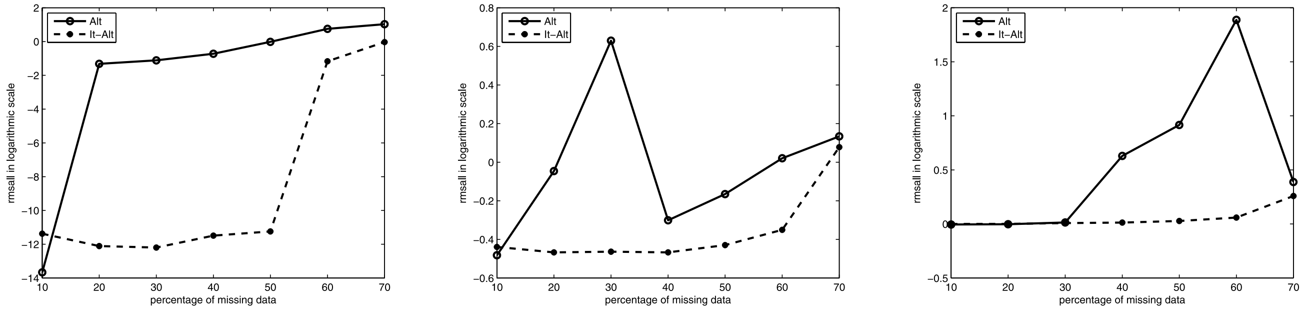


Fig. 14 Beethoven scene; mean of the rms_{all} in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

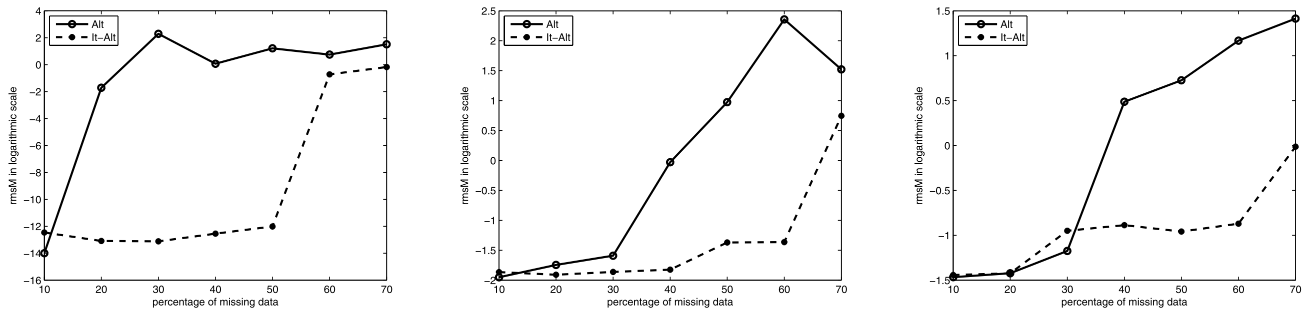


Fig. 15 Beethoven scene; mean of the error for the recovered motion (rms_M) in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

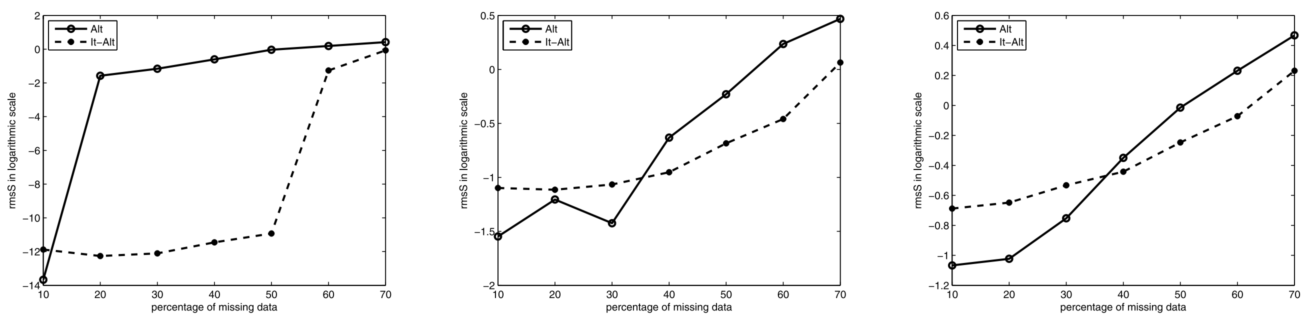


Fig. 16 Beethoven scene: mean of the error of the recovered shape (rms_S) in logarithmic scale, for different percentages of missing data: (*left*) no noise; (*middle*) $\sigma = 1/3$; (*right*) $\sigma = 1$

data case (Fig. 16 (left)), 30% for $\sigma = 1/3$ (Fig. 16 (middle)) and $\sigma = 1$ (Fig. 16 (right)).

As a conclusion, it can be seen that in general, the Alternation applied to the originally given input matrix performs quite well, while the percentage of missing data is small. Therefore, it is not worth 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 originally given input matrix converges to a local minimum increases as the percentage of missing data grows. See, for instance, how the mean of the *rms* varies between 20% and 30% 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 the originally given input matrix W. The reported results show that the Alternation applied to this partially or totally filled in matrix gives better results than when it is applied to the originally given input matrix.

3.2.2 *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

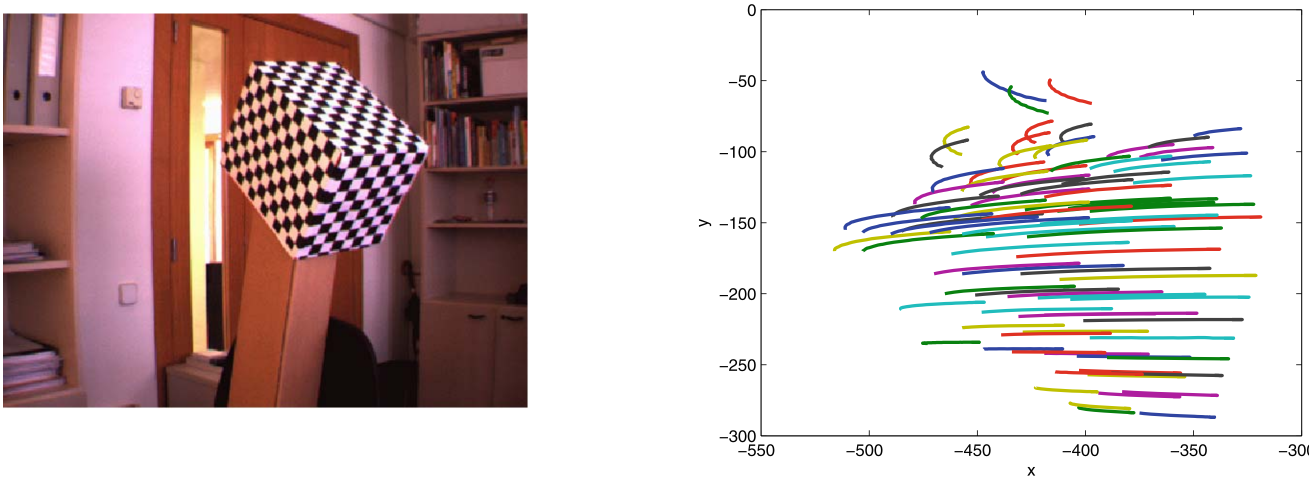


Fig. 17 (*left*) First object used for the real scene; (*right*) feature point trajectories represented in the image plane

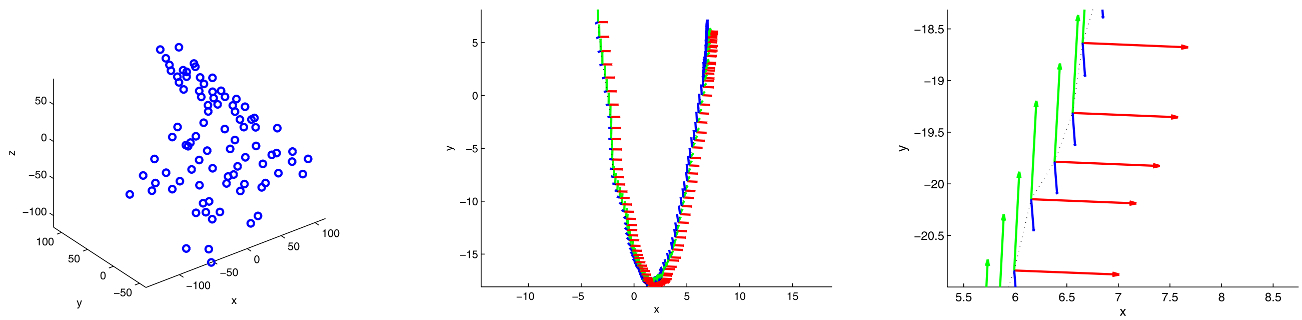


Fig. 18 First object; (*left*) 3D reconstruction; (*middle*) recovered camera motion; (*right*) enlargement of the recovered camera motion

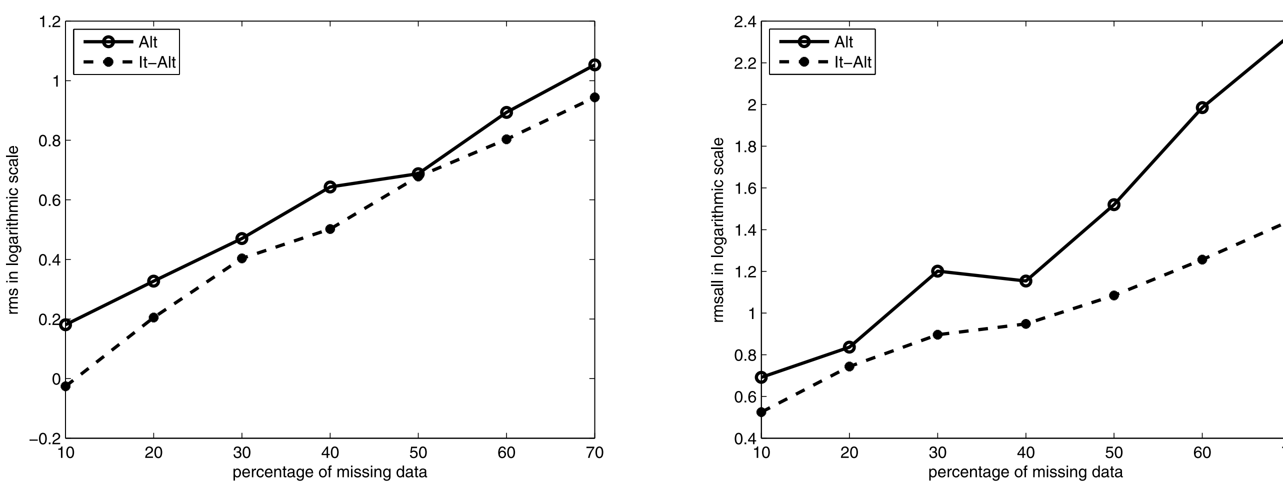


Fig. 19 First object: (*left*) mean of the *rms* in logarithmic scale, for different percentages of missing data; (*right*) mean of the rms_{all} in logarithmic scale, for different percentages of missing data

real data experiments are shown in Fig. 17 (left) and Fig. 20 (left). For each object, a video sequence with a resolution of 640 × 480 pixels is recorded and a single rotation around a vertical axis is performed. Feature points are selected by means of a corner detector algorithm proposed in [19] (Chap. 11, pp. 378–380). Concretely, features are selected by using the Harris' corner detector [10]. First of all, the gradient of the image is computed. Then, the quality of the fea-

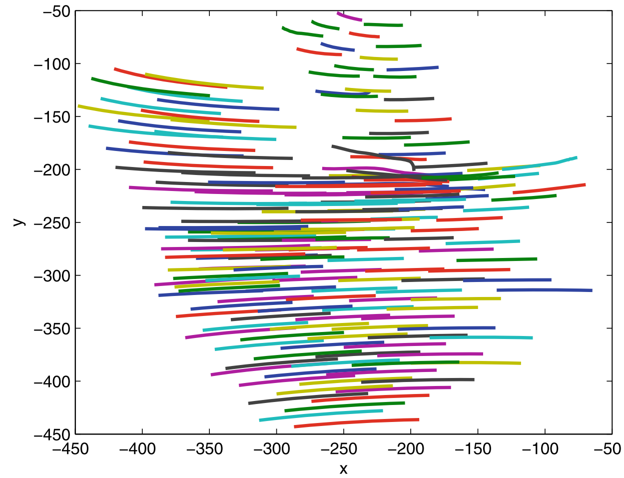


Fig. 20 (*left*) Second object used for the real scene; (*right*) feature point trajectories represented in the image plane

tures is measured by the Harris' criterion (see [19], Chap. 4, p. 91). Only features with a measure of quality higher than a given threshold are considered (in the current experiments the threshold has been set to 0.01). Next, the image is split into tiles, in order to obtain a more uniform feature selection. Finally, features are selected in each tile in decreasing order of quality (a maximum of 350 features per tile are selected in our experiments). Once the features are selected, the iterative feature tracking algorithm presented [19] (Chap. 11, pp. 380–390) is used. Feature points are tracked by minimizing the sum of squared differences between two consecutive frames.

As in the previous case, all the points are initially known in W_{all}, because only full trajectories are considered. Hence, during the experiments, different matrices containing missing data are automatically generated by removing parts of random columns (see Sect. 3.1), 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. 17 (right) and Fig. 20 (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. 17 (right). Figure 18 shows an example of the recovered shape (left) and motion ((middle) and (right)) obtained by applying Alternation to the matrix previously filled in with the proposed iterative scheme. In this example, the originally given input matrix contains only about 10% of missing data (recall that this 10% of missing data has been removed from the initially known trajectory matrix: W_{all}).

The resulting *rms* obtained for different percentages of missing data are presented in Fig. 19 (left). It can be seen that the Alternation applied to the matrix previously filled in with the iterative scheme performs better than applied directly to the originally given 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. 19 (right). Again, this error can be computed due to the fact that, in these particular experiments, the matrix W_{all} is initially known. As in the *rms*, the rms_{all} is smaller when the proposed scheme is previously used to fill in the given matrix; both values have been computed as introduced in Sect. 3.1.

The second sequence consists of 61 frames and 188 feature points. Feature point trajectories are plotted in Fig. 20 (right). Figure 21 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 originally given input matrix contains only about 10% of missing data.

In this second object, the error values are higher than in the first object, as it can be seen in Fig. 22. The *rms* (left) and the rms_{all} (right) are smaller when the Alternation is applied to the matrix filled in with the proposed iterative scheme than when applied to the originally given input one, for any percentage 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 when it is directly applied to the originally given input matrix, even when the percentage of missing data is low.

3.3 Iterative Multiresolution Scheme by Using Different Factorization Techniques

This section contains experimental results obtained when Powerfactorization [11] and the Alternation/Damped Newton Hybrid [2] are considered in the iterative multiresolution scheme and also at the final step, to factorize the resulting whole matrix. Therefore, results obtained when these two

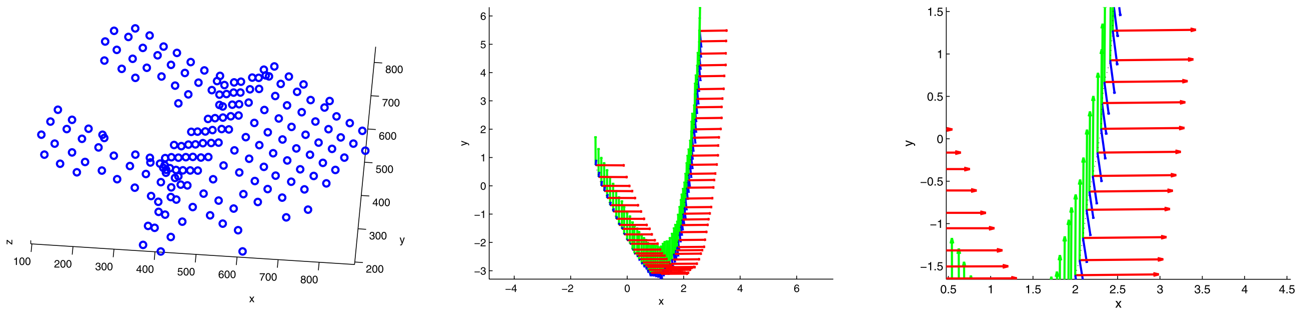


Fig. 21 Second object: (*left*) 3D reconstruction; (*middle*) recovered camera motion; (*right*) enlargement of the recovered camera motion

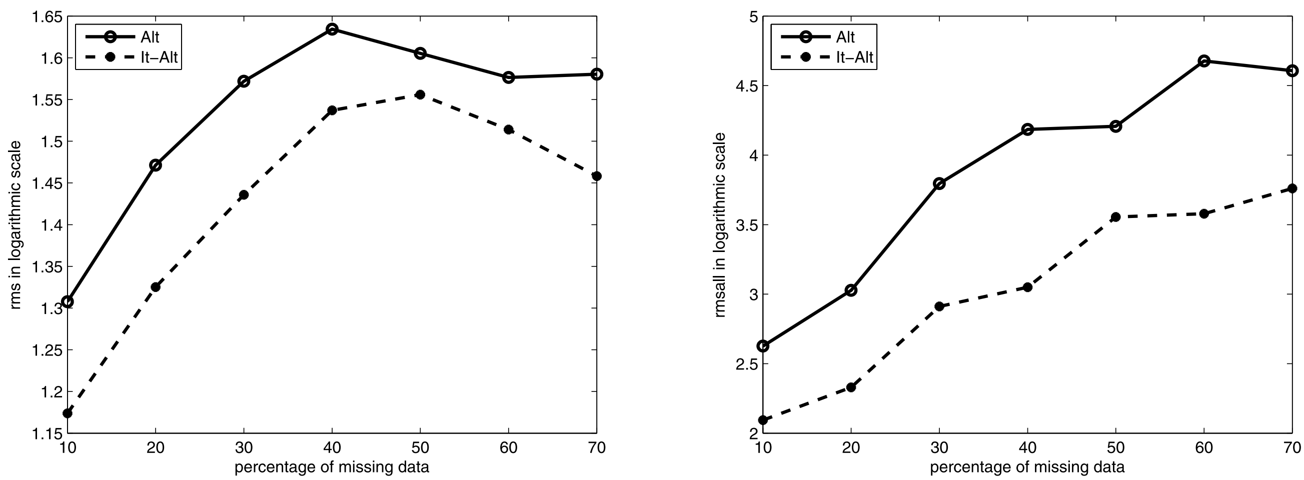


Fig. 22 Second object: (*left*) mean of the *rms* in logarithmic scale, for different percentages of missing data; (*right*) mean of the rms_{all} in logarithmic scale, for different percentages of missing data

factorization techniques are applied to an originally given input matrix, which can contain a high percentage of missing data, are compared to the results obtained when these techniques are applied to the corresponding matrix previously filled in with the proposed scheme.

Since the main objective of including this section is to show that the proposed scheme is also valid with other factorization techniques, only experiments with a synthetic object are provided. Concretely, the object presented in Fig. 11 is considered.

3.3.1 Powerfactorization

The Powerfactorization [11] technique is a variant of the Alternation that consists in adding a normalization step at each iteration. Concretely, Hartley et al. [11] propose to normalize the rows of the second factor (S in this particular problem). The idea of the algorithm is similar to the one of the Alternation; that is, factors are computed alternatively until its product converge to the initial matrix W. Therefore, computational cost is similar with both techniques.

Results obtained by using the Powerfactorization technique are shown in Fig. 23. It can be seen that results are improved when the Powerfactorization is applied to the matrix filled in with the proposed iterative scheme (denoted as It-PF), when the percentage of missing data is higher than 20% (see dashed line in the plot).

3.3.2 Alternation/Damped Newton Hybrid

Buchanan et al. [2] present the Damped Newton method for missing data matrix factorization. This method aims at minimizing an error function by performing a gradient descent step, which is updated at each iteration. However, authors point out that although this method is very fast when it is close to the solution, it cannot be effective when the initialization is far from minima. On the other hand, they also point out that the Alternation technique is initially fast but it can get stuck in local minimum, when the percentage of missing entries is very high.

In order to get advantage of both techniques, Buchanan et al. propose an hybrid method that combines them: the Alter-

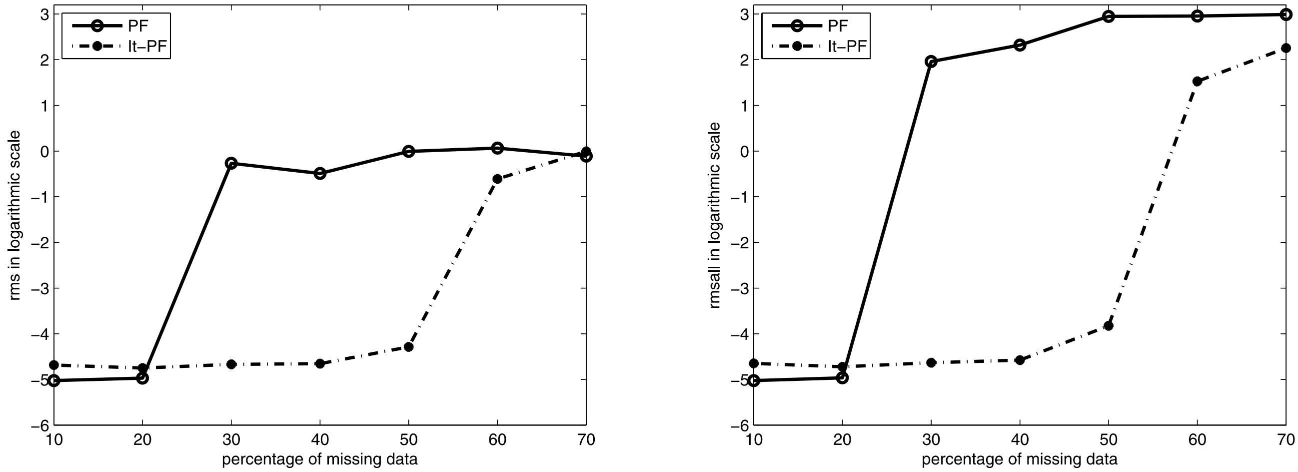


Fig. 23 Powerfactorization technique; mean of *rms* in logarithmic scale, for different percentages of missing data: (*left*) *rms*, considering only initially known entries; (*right*) rms_{all}, considering all entries in the originally given input matrix

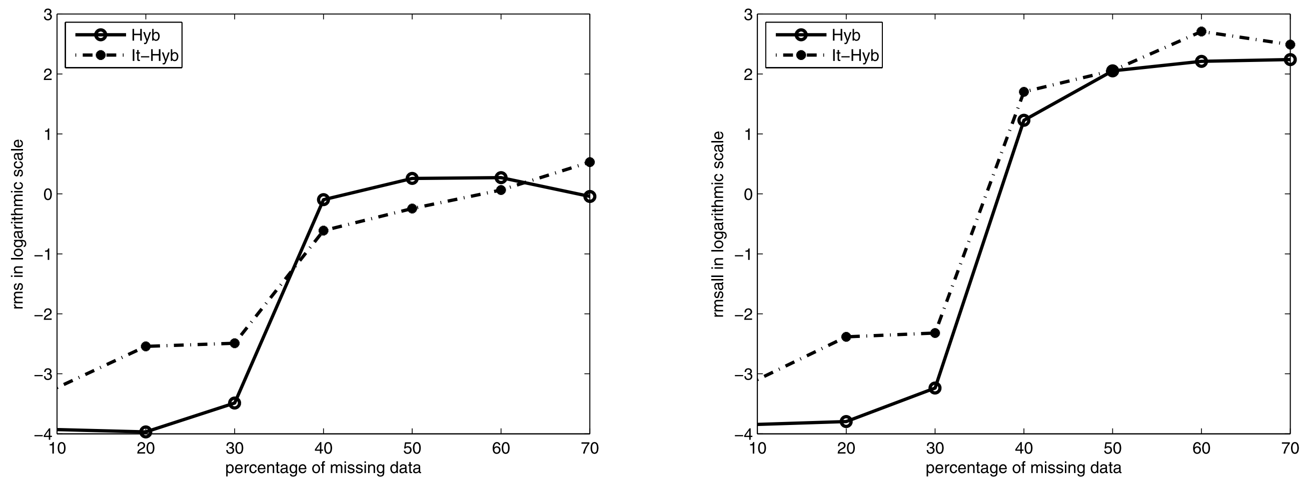


Fig. 24 Hybrid technique; mean of *rms* in logarithmic scale, for different percentages of missing data: (*left*) *rms*, considering only initially known entries; (*right*) rms_{all}, considering all entries in the original matrix

nation/Damped Newton Hybrid. Hence, this Hybrid method has an initial convergence and, at the same time, has the power of non-linear optimization. The key point of this Hybrid method is to decide when to switch from one method to the other. In the current paper, the third strategy proposed in [2] is used, which consists in performing a set of Alternation steps and then goes into the Hybrid scheme. In each iteration, if the parameter that controls step length and gradient-descent similarity is too large or worse than the last iteration, a different number of Alternation steps are performed.

Figure 24 shows the results obtained by applying the Hybrid technique to the originally given input matrix (denoted as Hyb) and to the matrix filled in with the proposed iterative scheme (denoted as It-Hyb). Recall that the Hybrid technique is also used inside the iterative scheme to fill in missing entries. It can be seen that, by using the Hybrid method, the improvements on the results when the iterative scheme is previously used are not significant (see for instance the cases 40, 50 and 60% of missing data in Fig. 24 (left)). Actually, similar results are obtained when the Hybrid method is directly applied to the originally given input matrix and to the filled in with the proposed iterative scheme. This is due to the fact that, in the Hybrid method, the number of known entries in the matrix W is not as important as in the Alternation and Powerfactorization techniques. Therefore, it might be expected that no big improvements would be obtained when the initial matrix was filled in with the proposed iterative scheme. Hybrid method has been used with the proposed scheme just to show that even in this case the proposed approach can provide similar, or slightly better results.

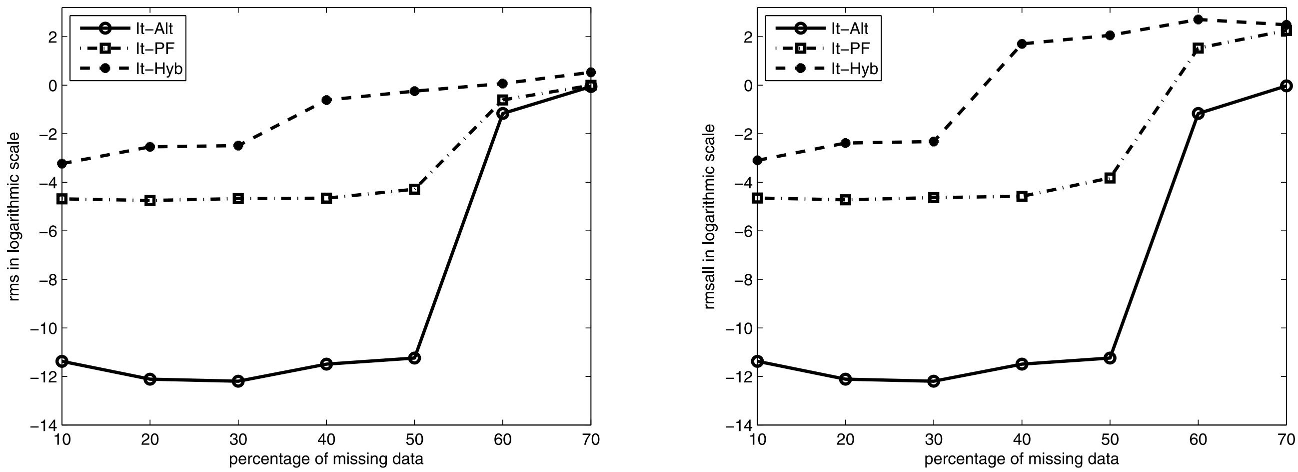


Fig. 25 Summary results obtained with the iterative scheme, considering different factorization techniques; (*left*) *rms*; (*right*) rms_{all}

3.4 Summary

Finally, Fig. 25 shows the results obtained with the iterative scheme, when the three different factorization techniques presented above are considered. The smallest error for this particular problem is obtained when the Alternation technique is considered in the proposed iterative multiresolution scheme.

The Hybrid method has a high computational cost, which makes it less suitable than the Alternation to be used inside the proposed iterative multiresolution scheme. In particular, the computational cost of the Hybrid method is about 100 times more expensive than the one of Alternation or Powerfactorization. Consequently, when it is applied inside the proposed iterative scheme (It-Hyb), the computational cost of the scheme is even higher, depending on the number of processed matrices. For instance, in the case of 20% of missing data (there is a high number of submatrices to be processed), the It-Hyb scheme takes about 10^7 times more than It-Alt or It-PF; while in the case of 50% of missing data It-Hyb takes about 10^3 times more than It-Alt or It-PF.

4 Conclusions and Future Work

This paper 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 iterative multiresolution scheme is to take sub-matrices of the input matrix with a low percentage of missing data, apply a factorization technique to obtain the shape S and motion M and hence recover the missing entries with the product MS. The goal is to improve the results obtained when a factorization technique is applied to the matrix filled in with this iterative scheme instead of directly to the originally given input one, which contains a lower percentage of known data.

An evaluation study of the performance of the proposed scheme is done. Although the study is focused on the Alternation technique, results obtained with different factorization techniques are provided. In particular, the Powerfactorization and the Alternation/Damped Newton Hybrid are considered, both to factorize the sub-matrices and to recover the motion and shape of the whole matrix at the final step. It has been shown that, when the percentage of missing data is high, the Alternation and Powerfactorization techniques applied to the matrix filled in with the proposed iterative scheme gives better results than when applied directly to the originally given input matrix W. However, when the ratio of missing data is low, every studied factorization technique performs quite well directly applied to the originally given input matrix directly and it is not necessary to use the iterative scheme. In the case of the Hybrid technique, similar results are obtained with both strategies. 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 experiments with synthetic data, recovered shape and motion matrices are also studied. In conclusion, 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.

As a future work, we would like to extend the iterative multiresolution scheme to handle scenes with multiple objects. Furthermore, it would be interesting to use the iterative scheme in applications different from the SFM.

Acknowledgements This work has been partially supported by the Government of Spain under projects TRA2007-62526/AUT and DPI2007-66556-C03-03, and research programme Consolider-Ingenio 2010: MIPRCV (CSD2007-00018).

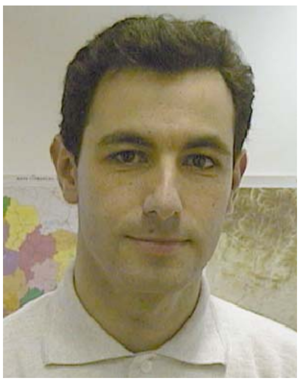
References

1. Aanaes, H., Fisker, R.: Robust factorization. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 1215–1225 (2002)
2. Buchanan, A., Fitzgibbon, A.W.: Damped Newton algorithms for matrix factorization with missing data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 316–322. University of Oxford, Oxford (2005)
3. Chen, P., Suter, D.: Recovering the missing components in a large noisy low-rank matrix: Application to SFM. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 1051–1063 (2004)
4. Costeira, J.P., Kanade, T.: A multibody factorization method for independently moving objects. Int. J. Comput. Vis. **3**, 159–179 (1998)
5. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Graph. Image Process. **24**, 381–395 (1981)
6. Golub, G.H., Van Loan, C.F. (eds.): Matrix Computations. Johns Hopkins Univ. Press, Baltimore (1989)
7. Guerreiro, R.F.C., Aguiar, P.M.Q.: Estimation of rank deficient matrices from partial observations: two-step iterative algorithms. In: EMMCVPR, pp. 450–466 (2003)
8. Guilbert, N., Bartoli, A., Heyden, A.: Affine approximation for direct batch recovery of Euclidean structure and motion from sparse data. Int. J. Comput. Vis. **69**, 317–333 (2006)
9. Han, M., Kanade, T.: Reconstruction of a scene with multiple linearly moving objects. Int. J. Comput. Vis. **53**, 285–300 (2000)
10. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Fourth Alvey Vision Conference, pp. 147–151 (1988)
11. Hartley, R., Schaffalitzky, F.: Powerfactorization: 3D reconstruction with missing or uncertain data. In: Australian-Japan Advanced Workshop on Computer Vision (2003)
12. Hartley, R., Zisserman, A. (eds.): Multiple View Geometry in Computer Vision. Cambridge University Press. Book web page at: http://www.robots.ox.ac.uk/~vgg/hzbook/index.html, accessed February 2007 (2000)
13. Huynh, D.Q., Hartley, R., Heyden, A.: Outlier correction in image sequences for the affine camera. In: IEEE ICCV, vol. 1, pp. 585–590 (2003)
14. Huynh, D.Q., Heyden, A.: Outlier detection in video sequences under affine projection. In: IEEE CVPR, vol. 1, pp. 695–701 (2001)
15. Jacobs, D.W.: Linear fitting with missing data for structure-from-motion. Comput. Vis. Image Underst. **1**, 57–81 (2001)
16. Jia, H., Fortuna, J., Martinez, A.: 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, pp. 1101–1007 (2006)
17. Julià, C., Sappa, A., Lumbreras, F., Serrat, J., López, A.: Factorization with missing and noisy data. In: Computational Science ICCS 2006: 6th International Conference. LNCS, vol. 1, pp. 555–562 (2006)
18. Julià, C., Sappa, A., Lumbreras, F., Serrat, J., López, A.: An iterative multiresolution scheme for SFM. In: ICIAR 2006. LNCS, vol. 4141, pp. 804–815 (2006)
19. Ma, Y., Soatto, J., Kosecká, J., Sastry, S.: An Invitation to 3D Vision: From Images to Geometric Models. Springer, New York (2004)
20. Martinec, D., Pajdla, T.: 3D reconstruction by fitting low-rank matrices with missing data. In: IEEE CVPR, vol. 1, pp. 198–205 (2005)
21. Morita, T., Kanade, T.: A sequential factorization method for recovering shape and motion from image streams. IEEE Trans. Pattern Anal. Mach. Intell. **19**(8), 858–867 (1997)
22. Morris, D., Kanade, T.: A unified factorization algorithm for points, line segments and planes with uncertainty models. In: International Conference on Computer Vision, pp. 696–702 (1998)
23. Okatani, T., Deguchi, K.: On the Wiberg algorithm for matrix factorization in the presence of missing components. Int. J. Comput. Vis. **72**, 329–337 (2007)
24. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 206–218 (1997)
25. Quan, L., Kanade, T.: A factorization method for affine structure from line correspondences. In: CVPR (1996)
26. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vis. **9**(2), 137–154 (1992)
27. Triggs, B.: Linear projective reconstruction from matching tensors. Image Vis. Comput. **15**, 617–625 (1997)
28. Wiberg, T.: Computation of principal components when data is missing. In: Second Symposium of Computational Statistics, pp. 229–326 (1976)

Carme Julià received the BSc degree in Mathematics in 2002 from the Polytechnic University of Catalonia, Barcelona, Spain. In October 2002 she started her PhD studies in Computer Science in the Computer Vision Center, at the Universitat Autònoma de Barcelona. In September 2004 she got the MSc degree and in 2008 the PhD degree with the thesis entitled Missing Data Matrix Factorization Addressing the Structure from Motion Problem. Since September 2008 she is a member of the Intelligent Robotics and Computer Vision Group, in the Universitat Rovira i Virgili, Tarragona, Spain. Her research interest is focused on Structure from Motion through factorization. Her current research is focused on factorizing matrices with missing and noisy data and on photometric stereo through factorization.

Angel D. Sappa received the electromechanical engineering degree from National University of La Pampa, General Pico, Argentina, in 1995 and the Ph.D. degree in industrial engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999. In 2003, after holding research positions in France, the UK, and Greece, he joined the Computer Vision Center, Barcelona, Spain, where he is currently a member of the Advanced Driver Assistance Systems Group. His research interests span a broad spectrum within the 2D and 3D image processing. His current research focuses on stereo image processing and analysis, 3D modelling, and model-based segmentation.

Felipe Lumbreras received the electromechanical BSc degree in physics from the Universitat de Barcelona in 1991 and the PhD degree in computer science from the Universitat Autònoma de Barcelona in 2001. He is currently an associated professor in the Computer Science Department and member of the Centre de Visió per Computador (CVC). His research interest include wavelet and texture analysis, 3D reconstruction, and computer vision for automotive applications.



Joan Serrat is associate lecturer at the computer science department of the Universitat Autònoma de Barcelona, and member of the Computer Vision Center. His areas of interest are structure from motion, 3D reconstruction and computer vision applied to driving assistance systems. He has also been head of several machine vision projects for local industries. He is member of the board of the IAPR Spanish chapter.

Antonio López received the B.Sc. degree in computer science from the Polytechnic University of Catalonia, Barcelona, Spain, in 1992 and the M.Sc. degree in image processing and artificial intelligence and the Ph.D. degree from the Universitat Autònoma de Barcelona (UAB) in 1994 and 2000, respectively. Since 1992, he has been giving lectures at the Computer Science Department, UAB, where he is currently an Associate Professor. In 1996, he participated in the foundation of the Computer Vision Center, UAB, where he has held different institutional responsibilities and is presently responsible for the research group on advanced driver-assistance systems by computer vision. He has been responsible for public and private projects. He has coauthored more than 50 papers, all in the field of computer vision.