Factorization with Missing and Noisy Data

Carme Julià, Angel Sappa, Felipe Lumbreras, Joan Serrat, and Antonio López

Computer Vision Center and Computer Science Department,
Universitat Autònoma de Barcelona,
08193 Bellaterra, Spain
{cjulia, asappa, felipe, joans, antonio}@cvc.uab.es

Abstract. Several factorization techniques have been proposed for tackling the *Structure from Motion* problem. Most of them provide a good solution, while the amount of missing data is within an acceptable ratio. Focussing on this problem, we propose an incremental multiresolution scheme able to deal with a high rate of missing data, as well as noisy data. It is based on an iterative approach that applies a classical factorization technique in an incrementally reduced space. Information recovered following a coarse-to-fine strategy is used for both, filling in the missing entries of the input matrix and denoising original data. A statistical study of the proposed scheme compared to a classical factorization technique is given. Experimental results obtained with synthetic data and real video sequences are presented to demonstrate the viability of the proposed approach.[1]

1   Introduction

Structure From Motion (SFM) consists in extracting the 3D shape of a scene as well as the camera motion from trajectories of tracked features. Factorization is a method addressing to this problem. The central idea is to express a matrix of trajectories W as the product of two unknown matrices, namely, the 3D object's shape S and the relative camera pose at each frame M: $W_{2f \times p} = M_{2f \times r} S_{r \times p}$, where f and p are the number of frames and feature points respectively and r the rank of $W_{2f \times p}$. These factors can be estimated thanks to the key result that their rank is small and due to constraints derived from the orthonormality of the camera axes. The *Singular Value Decomposition* (SVD) is generally used when there are not missing entries. Unfortunately, in most of the real cases not all the data points are available, hence other methods need to be used.

In the seminal approach Tomasi and Kanade [1] propose an initialization method in which they first decompose the largest full submatrix by the factorization method and then the initial solution grows by one row or by one column at a time, unveiling missing data. The problem is that finding the largest full submatrix of a matrix with missing entries is a NP-hard problem. Jacobs [2]

---

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. Missing entries are recovered by finding the least squares regression onto that subspace. However, this approach is strongly affected by noise on the data. An incremental SVD scheme of incomplete data is proposed by Brand [3]. The main drawback of that scheme is that the final result depends on the order in which the data are encountered. Brandt [4] proposes a different technique that addresses the affine reconstruction under missing data by means of an EM algorithm. Although the feature points do not have to be visible in all views, the affine projection matrices in each image must be known. A method for recovering the most reliable imputation, addressing the SFM problem, is provided by Suter and Chen [5]. They propose an iterative algorithm to employ this criterion to the problem of missing data. Their aim is not to obtain the factors M and S, but the projection onto a low rank matrix to reduce noise and to fill in missing data. Wiberg [6] introduces the *Alternation* technique to solve the factorization with missing data. Since then, several variants of this approach have been proposed in the literature. In [7], Buchanan and Fitzgibbon summarize different factorization approaches with missing data and propose the *Alternation/Damped Newton Hybrid*, which combines the *Alternation* strategy with the *Damped Newton* method.

One disadvantage of the above methods is that the result depends on the percentage of missing data. They give a good factorization while the amount of missing points is reduced, which is not common in real image sequences, unfortunately. Additionally to this problem, when real sequences are considered, the presence of noisy data needs to be taken into account in order to evaluate the performance of the factorization technique. Addressing to these problems, we propose to use an iterative multiresolution scheme, which incrementally fill in missing data. A statistical study of the performance of the proposed scheme is carried out considering different percentages of missing and noisy data. The key point of the implemented approach is to work with a reduced set of feature points along a few number of consecutive frames. Thus, the 3D reconstruction corresponding to the selected feature points and the camera motion of the used frames are obtained. Missing entries of the trajectory matrix are recovered, while, at the same time, noisy data are filtered.

This paper is organized as follows. Section 2 contains a brief review of Alternation factorization techniques for the case where there are missing data. Section 3 presents the incremental multiresolution scheme used to factorize a matrix of trajectories that has a large amount of missing data. The error function is defined in section 4. Section 5 contains results obtained with synthetic and real data. Conclusions and future work are given in section 6.

2   Alternation Technique

Let be $W_{2f \times p}$ the matrix of trajectories of p feature points, tracked over f frames—also denoted as W, or input matrix. The goal of Alternation is to find

the best rank r approximation to W, where $r < 2f, p$. That is, to compute the matrix factors M and S such that minimize the cost function:

$$\|W - MS\|_F^2 \tag{1}$$

where $\| \cdot \|$ is the Frobenius matrix norm [8]. In the case of missing data:

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

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

The algorithm starts with an initial random $2f \times r$ matrix M_0 and repeats the next steps until the product $M_k S_k$ converges to W:

$$S_k = (M_{k-1}^t M_{k-1})^{-1} M_{k-1}^t W \qquad M_k = W S_k (S_k^t S_k)^{-1} \tag{3}$$

As pointed out in [9], the most important advantage of this 2-step algorithm is that these equations are the matrix versions of the normal equations. That is, each M_k and S_k is the least-squares solution of a set of equations of the form $W = MS$. Besides, since the updates of M given S (and analogously in the case of S given M) can be independently done for each row of S, missing entries in W correspond to omitted equations. Due to that fact, with a few data points the method would fail to converge, but this happens only with large amounts of missing data.

In the application of affine SFM, the last row of S should be filled with ones $S = \begin{bmatrix} X & \mathbf{1} \end{bmatrix}^t$, where X are the 3D recovered coordinates of the feature points. Through the paper, an Alternation for SFM with motion constraints (AM) approach has been used. Hence, at each iteration k, it is used the fact that M is the motion matrix—the relative camera pose at each frame. Therefore, given M_{k-1}, and before computing S_k, we impose the orthonormality of the camera axes at each frame.

3   Proposed Approach

We propose an iterative multiresolution approach using the AM previously described. We will refer it as Incremental Alternation with Motion constraints (IAM). Essentially, our basic idea is to generate sub-matrices with a reduced density of missing points. Thus, the AM could be used for factoring these sub-matrices and recovering their corresponding 3D shape and motion. The proposed technique consists of two stages, which are fully explained below.

3.1   Observation Matrix Splitting

Let $W_{2f \times p}$ be the observation matrix of p feature points tracked over f frames containing missing entries. Let k be the index denoting the current iteration number. In a first step, W is split into $k \times k$ non-overlapped sub-matrices, each

one of them defined as $W_{k(i,j)}$, $i \in (0, \lfloor \frac{2f}{k} \rfloor]$, $j \in (0, \lfloor \frac{p}{k} \rfloor]$. For the sake of presentation simplicity, hereinafter a sub-matrix in the current iteration level k will be referred as W_k (assuming $k > 1$, since $k = 1$ is simply the AM method).

Although the idea is to focus the process in a small area (sub-matrix W_k), with a reduced density of missing data, recovering information from a small patch can be easily affected from noisy data, as pointed out in [5]. Hence there is a trade off between the size of a sub-matrix and the confidence of its recovered data. In order to improve the confidence of recovered data a multiresolution approach is followed. In a second step, and only when $k > 2$, four W_{2k} overlapped sub-matrices, with twice the size of W_k are computed as illustrated in Fig.1. The idea of this enlargement process is to study the behavior of feature points contained in W_k when a bigger region is considered. Other strategies were tested in order to compute in a fast and robust way sub-matrices with a reduced density of missing entries (e.g. quadtrees, ternary graph structure), but they do not give the desired and necessary properties of overlapping.

Since generating four W_{2k} for every W_k is a computationally expensive task, a simple and more direct approach is followed. It consists in splitting the input matrix W in four different ways, by shifting W_{2k} half of its size (i.e., W_k) through rows, columns or both at the same time. When all these matrices are considered together, the overlap between the different areas is obtained.
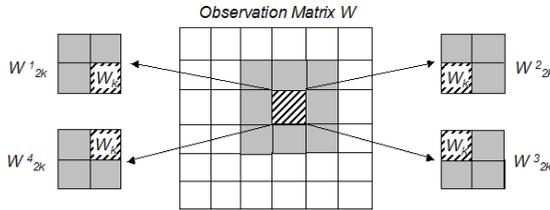

Fig. 1. W_{2k} overlapped matrices of the observation matrix W, computed during the first stage (section 3.1), at iteration $k = 6$

3.2   Sub-matrices Processing

At this stage, the objective is to recover missing data by applying AM at every single sub-matrix. Independently of their size hereinafter sub-matrices will be referred as W_i.

Given a sub-matrix W_i, the AM gives its corresponding M_i and S_i matrices. Their product could be used for computing an approximation error ε_i such as equation (2). In case the resulting error is smaller than a user defined threshold σ, every point in W_i is kept in order to be merged with overlapped values after finishing the current iteration. Additionally, every point of W_i is associated with a weighting factor, defined as $\frac{1}{\varepsilon_i}$, in order to measure the goodness of that value. These weighting factors are later on used for merging data on overlapped areas. Otherwise, the resulting error is higher than σ, computed data are discarded.

Finally, when every sub-matrix W_i has been processed, recovered missing data are used for filling in the input matrix W. In case a missing datum has been recovered from more than one sub-matrix (overlapped regions), those recovered data are merged by using their corresponding normalized weighting factors. On the contrary, when a missing datum has been recovered from only one sub-matrix, this value is directly used for filling in that position.

Once recovered missing data were used for filling in the input matrix W, the iterative process starts again (section 3.1) splitting the new matrix W (the input one merged with recovered data) either by incrementing k one unit or, in case the size of sub-matrices W_k at the new iteration stage is quite small (the smaller W_k size was set to 5×5), by setting $k = 2$. This iterative process is applied until one of the following conditions is true: a) the matrix of trajectories is totally filled; b) at the current iteration no missing data were recovered; c) a maximum number of iterations is reached.

4   The Error Function

As pointed out in [5], the cost function defined by equation (2) could be ambiguous and in some cases contradictory. That is because that formula only takes into account the recovered values corresponding to known features, but it ignores how the rest of entries are filled. Therefore, in order to compare the proposed IAM scheme with the classical AM, an error function that considers all the features of the sequence, including missing points, is used (see equation (1)). Unfortunately, this is only possible when we have access to the whole information. Hence, when real data are considered, in order to perform a comparison by using the proposed error function, a full matrix should be selected. This matrix is used as input and missing data are randomly removed.

As it will be presented in short, noise is added to the input matrix W. However, since the aim is also to study how the data are filtered, the entries of the matrix of trajectories given by the product MS are compared with the corresponding elements in the input matrix W.

5   Experimental Results

As it was previously mentioned we want to do an statistical study about the filtering capability of the IAM scheme compared to the classical AM. At the same time, the robustness to missing data will be considered. In order to perform a comparison that help us to infer some conclusions, different levels of Gaussian noise—standard deviation (also denoted as std) with values from $\frac{1}{8}$ to 1, both in a synthetic and a real case—are added into the 2D feature point trajectories and different amounts of missing data are considered—from 20% up to 80%. Statistical results are obtained by applying: a) AM over the input matrix W; b) AM after filling in missing data with the proposed IAM. Although this strategy consists of two parts (IAM+AM), for simplicity, we referred it as IAM.

For each setting (level of noise, amount of missing data) 100 attempts are repeated and the number of *convergent* cases—those in which the error value is smaller than a threshold—obtained with each approach is computed. Due to the fact that for each setting the error takes a different range of values, a unique threshold is difficult to define. Therefore, it is defined for each particular setting, by the mean of the inliers error values ϵ [2] obtained from AM and IAM. Notice that more *convergent* cases does not necessary mean that a better performance is achieved for that setting. The idea is compare AM and IAM, not the different settings. Experiments using both synthetic and real data are presented below.

5.1 Synthetic Object

Synthetic data are randomly generated by distributing 35 3D feature points over the whole surface of a cylinder, see Fig.2 (left). The cylinder is defined by a radius of 100 and a height of 400; it rotates over its axis. The corresponding input matrices are obtained using different number of frames. In order to obtain a low percentage of missing data, a few frames are taken and the resulting input matrices are quite small. The IAM performs worse in these cases. However, the goal is to show its performance for a great amount of missing data, when other factorization techniques tend to fail. Additionally, the small size of the input matrices does not help to see the denoising capability of both AM and IAM.
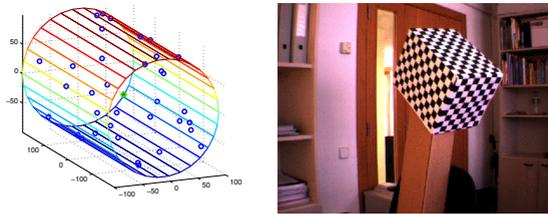

Fig. 2. (left) Synthetic object used to test the proposed approach. (right) Input object used for the real case.

As shown in Fig. 3 (left), the case of free-noisy data differs considerably from the others. In particular, for the IAM approach, a high average of convergent cases is obtained, no matter the ratio of missing data. Notice that with IAM and for 80% of missing data, a ratio of convergence of 100 is obtained. That does not mean a better result that working with a low percentage of missing data. As mentioned above, the idea is not to compare with the other settings, but with the AM, which has a ratio of convergence of about 2 for the same setting.

---

[2] The inliers ϵ are defined as $|\epsilon| < q_3 + 1.5 \Delta q$—where q_3 is the value of the third quartile and Δq is the interquartile distance $q_3 - q_1$.
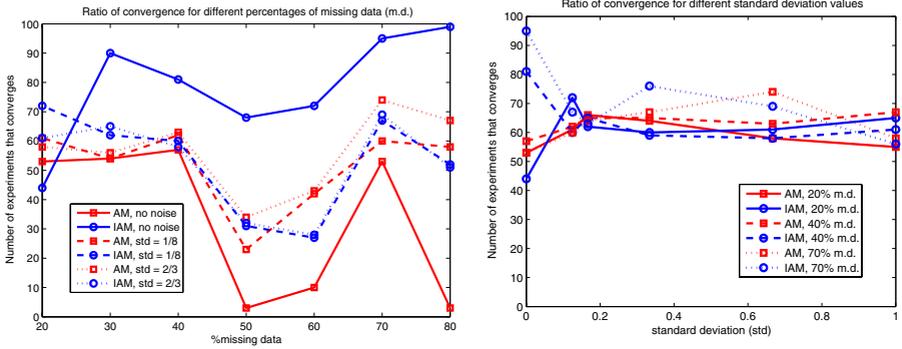
Fig. 3. Synthetic case. (left) Ratio of convergence for the different percentages of missing data, fixing different standard deviation (*std*) values. (right) The same for different *std* values, fixing various percentages of missing data. Zero *std* means no noisy data.

5.2 Real Object

Experimental results with a real video sequence of 101 frames with a resolution of 640×480 pixels are presented. The studied object is shown in Fig. 2 (right). A single rotation around a vertical axis is performed. Feature points are selected by means of a corner detector algorithm and 87 points over the object to be studied are considered. An iterative feature tracking algorithm has been used. More details about corner detection and tracking algorithm can be found in [10]. Different ratios of missing data are obtained by randomly removing data; the removed data are recorded in order to compute the error value (1).

Again, in Fig. 4 (left) it seems that for the case of no noise and a percentage of missing data from 40 up to 60, both AM and IAM performs worse than for other percentages. As shown in Fig. 4 (right), the number of convergent cases is in general higher applying IAM than AM.
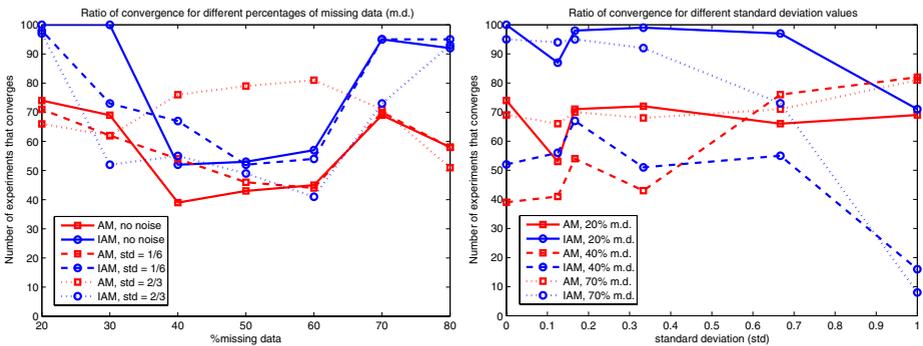

Fig. 4. Real case. (left) Ratio of convergence for the different percentages of missing data, fixing different standard deviation (*std*) values. (right) The same for different *std* values, fixing various percentages of missing data. Zero *std* means no noisy data.

6   Conclusions and Future Work

This paper presents an efficient technique for tackling the SFM problem when a high ratio of missing and noisy data is considered. The proposed approach exploits the simplicity of an Alternation technique by means of an iterative scheme. Missing data are incrementally recovered improving the final results. Noise have been added to the data and a statistical study about the filtering capability of AM compared to our incremental strategy have been done. It has been shown that, in most of the cases, results of IAM are better than the ones of AM in the sense of number of convergent cases.

In the future, we would like to use the proposed icremental multiresolution scheme with other classical factorization techniques. Additionally, other functions that consider the goodness of the obtained M and S and not only the recovered elements of W will be studied.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams: a factorization method. Full report on the orthographic case (1992)
2. Jacobs, D.: Linear fitting with missing data for structure-from-motion. Computer vision and image understanding, CVIU (2001) 7–81
3. Brand, M.: Incremental singular value decomposition of uncertain data with missing values. In: Proceedings, ECCV. (2002) 707–720
4. Brandt, S.: Closed-form solutions for affine reconstruction under missing data. In: Proceedings Statistical Methods for Video Processing Workshop, in conjunction with ECCV. (2002) 109–114
5. Chen, P., Suter, D.: Recovering the missing components in a large noisy low-rank matrix: Application to sfm. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004)
6. Wiberg, T.: Computation of principal components when data is missing. In: Proceedings Second Symposium of Computational Statistics. (1976) 229–326
7. Buchanan, A., Fitzgibbon, A.: Damped newton algorithms for matrix factorization with missing data. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) **2** (2005) 316–322
8. Golub, G., Van Loan, C., eds.: Matrix Computations. The Johns Hopkins Univ. Press (1989)
9. Hartley, R., Schaffalitzky, F.: Powerfactorization: 3d reconstruction with missing or uncertain data. Australian-Japan advanced workshop on Computer Vision (2003)
10. Ma, Y., Soatto, J., Koseck, J., Sastry, S.: An invitation to 3d vision: From images to geometric models. Springer-Verlang, New York (2004)