Lecture Notes in Computer Science 2683
International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition
July 2003, Lisbon, Portugal

Estimation of Rank Deficient Matrices from Partial Observations: Two-Step Iterative Algorithms *

Rui F. C. Guerreiro and Pedro M. Q. Aguiar

Institute for Systems and Robotics / IST, Lisboa, Portugal
{rfcg,aguiar}@isr.ist.utl.pt,
WWW home page: http://www.isr.ist.utl.pt/~{rfcg,aguiar}

Abstract. Several computer vision applications require estimating a rank deficient matrix from noisy observations of its entries. When the observation matrix has no missing data, the LS solution of such problem is known to be given by the SVD. However, in practice, when several entries of the matrix are not observed, the problem has no closed form solution. In this paper, we study two iterative algorithms for minimizing the non-linear LS cost function obtained when estimating rank deficient matrices from partial observations. In the first algorithm, the iterations are the well known *Expectation* and *Maximization* (EM) steps that have succeeded in several estimation problems with missing data. The second algorithm, which we call *Row-Column* (RC), estimates, in alternate steps, the row and column spaces of the solution matrix. Our conclusions are that RC performs better than EM in what respects to the robustness to the initialization and to the convergence speed. We also demonstrate the algorithms when inferring 3D structure from video sequences.

1 Introduction

Recent approaches to several computer vision problems require determining linear or affine low dimensional subspaces from noisy observations. These problems include object recognition [1, 2], applications in photometry [3–5], image alignment [6, 7], and the recovery of 3D rigid structure from video sequences [8–17].

In general, such low dimensional subspaces are found by estimating rank deficient matrices from noisy observations of their entries. When the observation matrix is completely known, the solution to this problem is easily obtained from its *Singular Value Decomposition* (SVD) [18]. However, in practice, the observation matrix may be incomplete, *i.e.*, some of its entries may be unknown (unobserved). Take as an example the recovery of 3D structure from video. The observation matrix collects 2D trajectories of projections of feature points [8–13] or other primitives [14–17]. In real life video clips, these projections are not visible along the entire image sequence due to the occlusion and the limited field of view. Thus, the observation matrix is in general incomplete.

* This work was partially supported by FCT project POSI/SRI/41561/2001, FCT POSI – QCA III.

In this paper, we address the problem of estimating a rank deficient matrix from noisy observations of a subset of its entries. This problem hasn't been much addressed in the computer vision literature. References [8] and [19] propose sub-optimal solutions. In [8], the missing values of the observation matrix are "filled in", in a sequential way, by using SVDs of observed submatrices. In [19], the author proposes a method to combine the constrains that arise from the observed submatrices of the original matrix. A bidirectional optimization scheme was proposed in [20]. In this paper, we study two distinct two-step iterative algorithms, developed for this problem. The first algorithm is based in a well known method to deal with missing data – the *Expectation-Maximization* (EM) [21]. Although the authors don't refer it, the bidirectional scheme in [20] is also an EM-based algorithm. However, as detailed below, the EM algorithm we propose is more general and computationally simpler than the one in [20]. Our second two-step iterative scheme is similar to Wiberg's algorithm [22] and related to the one used in [23] to model polyhedral objects. It computes, alternately, in closed form, the row space matrix and the column space matrix whose product is the solution matrix. We call this the *Row-Column* (RC) iterative algorithm.

In the paper, we illustrate the behavior of both algorithms with simple cases and evaluate their performance with more extensive experiments. In particular, we study the impact of the initialization on the algorithm's behavior. From these experiments, we conclude that the RC algorithm is more robust than EM in what respects to the sensibility to the initialization. Furthermore, the number of iterations needed for good convergence and the computational cost of each iteration are both smaller for RC than for EM. Obviously, the performances of both EM and RC improve when the initial estimate provided to the algorithms is more accurate. Any sub-optimal method, e.g. the ones in [8, 19], can be used to compute such an initialization. Our experience shows that, with a simple initialization procedure, even for high levels of noise and large amount of missing data, both iterative algorithms converge: **i)** to the global optimum; and **ii)** in a very small number of iterations.

We apply the EM and RC estimation algorithms to the problem of recovering 3D rigid structure from video sequences, when the observation matrix misses entries due to occlusion. Our experiments show that fitting the rank deficient matrix to the entire observation matrix (which misses several entries) leads to better 3D reconstructions than those obtained by combining partial 3D models estimated by fitting submatrices to smaller subsets of data (each corresponding to a subset of features that were visible in a subset of frames). A preliminary version of parts of this work was presented in [24]. MatLab© implementations of the algorithms we describe in this paper are available at [25].

Paper organization In Section 2, we introduce the *Least Squares* (LS) cost function associated with the problem of estimating a rank deficient matrix from noisy observations of a subset of its entries. Sections 3 and 4 describe the EM and RC iterative algorithms that minimize the non-linear LS cost function. In Section 5, we illustrate the behavior of the EM and RC algorithms with simple cases that enable graphical representations and demonstrate their good performance

when dealing with arbitrary matrices. In Section 6 we apply our algorithms to the problem of recovering 3D rigid structure from video sequences. Section 7 concludes the paper.

2 Problem Formulation

Given an observation $\boldsymbol{W}$ of a $M \times N$ rank deficient matrix $\widetilde{\boldsymbol{W}}$, say rank $R < \min(M, N)$, corrupted by white Gaussian noise, the ML estimate $\widehat{\boldsymbol{W}}$ of $\widetilde{\boldsymbol{W}}$ is

$$\widehat{\boldsymbol{W}} = \arg \min_{\widetilde{\boldsymbol{W}} \in \mathcal{S}_R} \left\| \boldsymbol{W} - \widetilde{\boldsymbol{W}} \right\|_F , \tag{1}$$

where $\|.\|_F$ represents the Frobenius norm and $\mathcal{S}_R$ denotes the space of the $M \times N$ rank R matrices. The solution $\widehat{\boldsymbol{W}}$ of (1) is known – it is obtained from the SVD of $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$, after selecting the R largest singular values [18]. We denote this optimal rank reduction operation, *i.e.*, the projection onto $\mathcal{S}_R$, by $\boldsymbol{W} \downarrow \mathcal{S}_R$:

$$\widehat{\boldsymbol{W}} = \boldsymbol{W} \downarrow \mathcal{S}_R = \boldsymbol{U}_{M \times R} \, \boldsymbol{\Sigma}_{R \times R} \, \boldsymbol{V}_{R \times N} . \tag{2}$$

When the observation matrix $\boldsymbol{W}$ misses a subset of its entries, the ML estimation of $\widetilde{\boldsymbol{W}}$ leads to the minimization of a generalized version of (1),

$$\widehat{\boldsymbol{W}} = \arg \min_{\widetilde{\boldsymbol{W}} \in \mathcal{S}_R} \left\| \left(\boldsymbol{W} - \widetilde{\boldsymbol{W}} \right) \odot \boldsymbol{M} \right\|_F , \tag{3}$$

where $\odot$ represents the elementwise product, also known as the Hadamard product, and the $M \times N$ matrix $\boldsymbol{M}$ is a binary mask that accounts for the known entries of the observation matrix $\boldsymbol{W}$, *i.e.*, $m_{ij} = 1$ if w_{ij} is known and $m_{ij} = 0$ otherwise. The existence of unknown entries in $\boldsymbol{W}$ prevents us to use the SVD of $\boldsymbol{W}$ as in (2) to minimize (3). In Sections 3 and 4 we introduce two iterative algorithms that minimize the nonlinear cost function (3).

3 Expectation-Maximization Algorithm

The *Expectation-Maximization* (EM) approach to estimation problems with missing data works by enlarging the set of parameters to estimate – the data that is missing is jointly estimated with the other parameters. The joint estimation is performed, iteratively, in two alternate steps: **i)** the *E-step* estimates the missing data given the previous estimate of the other parameters; **ii)** the *M-step* estimates the other parameters given the previous estimate of the missing data, see [21] for a review on the EM algorithm.

In our case, given an initial estimate $\widehat{\boldsymbol{W}}_0$, the EM algorithm estimates in alternate steps: **i)** the missing entries of the observation matrix $\boldsymbol{W}$; **ii)** the rank R matrix $\widehat{\boldsymbol{W}}$ that best matches the "complete" data. The algorithm performs these

two steps until convergence, *i.e.*, until the error measured by the Frobenius norm in (3) stabilizes.

E-step – estimation of the missing data Given $\widehat{\boldsymbol{W}}_{k-1}$, the ML estimates of the missing entries $\{w_{ij} : m_{ij}=0\}$ of $\boldsymbol{W}$ are simply the corresponding entries $\widehat{w}_{ij}$ of $\widehat{\boldsymbol{W}}_{k-1}$. We then build a complete observation matrix $\overline{\boldsymbol{W}}_k$, whose entry $\overline{w}_{ij}$ equals the corresponding entry w_{ij} of the observation matrix $\boldsymbol{W}$ if w_{ij} was observed or its estimate $\widehat{w}_{ij}$ if w_{ij} is unknown,

$$\overline{w}_{ij} = \begin{cases} w_{ij} \text{ if } \mathrm{m_{ij}} = 1 \\ \widehat{w}_{ij} \text{ if } \mathrm{m_{ij}} = 0, \end{cases} \tag{4}$$

or, in matrix notation,

$$\overline{\boldsymbol{W}}_k = \boldsymbol{W} \odot \boldsymbol{M} + \widehat{\boldsymbol{W}}_{k-1} \odot [\boldsymbol{1} - \boldsymbol{M}]\,. \tag{5}$$

M-step – estimation of the rank deficient matrix We are now given the complete data matrix $\overline{\boldsymbol{W}}_k$ with the estimates of the missing data from the E-step. The ML estimate of the rank R matrix $\widetilde{\boldsymbol{W}}$, *i.e.*, the rank R matrix $\widehat{\boldsymbol{W}}_k$ that best matches $\overline{\boldsymbol{W}}_k$ in the Frobenius norm sense, is then obtained from the SVD of $\overline{\boldsymbol{W}}_k$, as in (2),

$$\widehat{\boldsymbol{W}}_k = \overline{\boldsymbol{W}}_k \downarrow \mathcal{S}_R. \tag{6}$$

In [20], the authors develop a bidirectional algorithm to factor out an observation matrix with missing data, in the context of recovering rigid SFM. Their bidirectional algorithm is in fact an EM algorithm developed to the specific strategy of treating the 3D translation separately. In opposition, the EM algorithm just described is general, i.e, it solves any rank deficient matrix approximation problem with missing data. Furthermore, our E-step in (5) is simpler than the corresponding step of [20] that requires inverting matrices.

4 Row-Column Algorithm

We now describe the *Row-Column* (RC) algorithm – another iterative approach, similar to Wiberg's algorithm [22], to the estimation of a rank deficient matrix that best matches an incomplete observation. From our experience, summarized in Section 5, the RC algorithm is not only computationally cheaper than EM, avoiding SVD computations and exhibiting faster convergence, but also more robust than EM to initializations far from the solution.

For the RC algorithm, we parameterize the rank R matrix $\widetilde{\boldsymbol{W}}$ as the product

$$\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{A}}_{M\times R}\, \widetilde{\boldsymbol{B}}_{R\times N} \in \mathcal{S}_R, \tag{7}$$

where $\widetilde{\boldsymbol{A}}$ determines the column space of $\widetilde{\boldsymbol{W}}$ and $\widetilde{\boldsymbol{B}}$ its row space. The estimate $\widehat{\boldsymbol{W}}$ of $\widetilde{\boldsymbol{W}}$ is obtained by minimizing the cost function in (3) with respect to (wrt) the column space and row space matrices, *i.e.*, $\widehat{\boldsymbol{W}} = \boldsymbol{A}\boldsymbol{B}$, where

$$\{\boldsymbol{A}, \boldsymbol{B}\} = \arg\min_{\widetilde{\boldsymbol{A}},\widetilde{\boldsymbol{B}}} \left\| \left(\boldsymbol{W} - \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}}\right) \odot \boldsymbol{M} \right\|_F. \tag{8}$$

By using this parameterization, we have mapped the constrained minimization (3) wrt $\widetilde{\boldsymbol{W}} \in \mathcal{S}_R$ into the unconstrained minimization (8) wrt $\widetilde{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{B}}$.

We minimize (8) in two alternate steps: **i)** the *R-step* assumes the column space matrix $\boldsymbol{A}$ is known and estimates the row space matrix $\boldsymbol{B}$; **ii)** the *C-step* estimates $\boldsymbol{B}$ for known $\boldsymbol{A}$. The algorithm is initialized by computing $\boldsymbol{A}$ from an initial estimate $\widehat{\boldsymbol{W}}_0$ and it runs until the value of the norm in (8) stabilizes.

When there is no missing data, *i.e.*, when $\boldsymbol{M} = \mathbf{1}_{M \times N}$, the solutions for the RC steps above are simply obtained by using the *pseudoinverse* [18],

$$\boldsymbol{B}_k = \left(\boldsymbol{A}_{k-1}^T \boldsymbol{A}_{k-1}\right)^{-1} \boldsymbol{A}_{k-1}^T \boldsymbol{W}, \qquad \boldsymbol{A}_k = \boldsymbol{W} \boldsymbol{B}_k^T \left(\boldsymbol{B}_k \boldsymbol{B}_k^T\right)^{-1}. \tag{9}$$

If we write steps R and C together as a recursion on one of the matrices $\boldsymbol{A}$ or $\boldsymbol{B}$, say, on the column space in $\boldsymbol{A}$, we get

$$\boldsymbol{A}_k = \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{A}_{k-1} \left(\boldsymbol{A}_{k-1}^T \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{A}_{k-1}\right)^{-1} \boldsymbol{A}_{k-1}^T \boldsymbol{A}_{k-1}, \tag{10}$$

which shows that, in this simpler case, our RC algorithm is in fact implementing the application of the *power method* [18] to the matrix $\boldsymbol{W}\boldsymbol{W}^T$ (the factor $(\boldsymbol{A}_{k-1}^T \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{A}_{k-1})^{-1} \boldsymbol{A}_{k-1}^T \boldsymbol{A}_{k-1}$ is the normalization). The power method has been widely used to avoid the computation of the entire SVD when fitting rank deficient matrices to complete observations. We will see that, even when there is missing data, steps R and C admit closed-form solution and the overall algorithm generalizes the power method in a very simple way.

R-step – estimation of the row space For known $\boldsymbol{A}$, the minimization of (8) wrt $\boldsymbol{B}$ can be rewritten in terms of each of the N columns $\{\boldsymbol{b}_n, n = 1 \ldots N\}$ of $\boldsymbol{B}$,

$$\boldsymbol{b}_n = \arg\min_{\widetilde{\boldsymbol{b}}_n} \left\| \left(\boldsymbol{w}_n - \boldsymbol{A}\widetilde{\boldsymbol{b}}_n\right) \odot \boldsymbol{m}_n \right\|_F, \tag{11}$$

where the lowercase boldface letters denote columns of the matrices with the same uppercase letters. Exploiting the structure of the vector $\boldsymbol{m}_n$, we now rearrange the minimization in (11) in such a way that its solution becomes obvious. First, we note that the binary vector $\boldsymbol{m}_n$ in (11) is just selecting the entries of the error vector $(\boldsymbol{w}_n - \boldsymbol{A}\widetilde{\boldsymbol{b}}_n)$ that affect the error norm. Then, by making explicit that selection in terms of the entries of $\boldsymbol{w}_n$ that contain known data and the corresponding relevant entries of $\boldsymbol{A}$, we rewrite (11) as

$$\boldsymbol{b}_n = \arg\min_{\widetilde{\boldsymbol{b}}_n} \left\| \boldsymbol{w}_n \odot \boldsymbol{m}_n - (\boldsymbol{A} \odot \boldsymbol{M}_n)\, \widetilde{\boldsymbol{b}}_n \right\|_F, \tag{12}$$

where $\boldsymbol{M}_n$ is a $M \times R$ matrix with all R columns equal to $\boldsymbol{m}_n$, *i.e.*, it is a short notation for $\boldsymbol{M}_n = \boldsymbol{m}_n \mathbf{1}_{1 \times R}$.

The minimization in (12) is now a linear LS problem. Its solution $\boldsymbol{b}_n$ is then obtained by using the pseudoinverse of matrix $\boldsymbol{A} \odot \boldsymbol{M}_n$,

$$\boldsymbol{b}_n = \left[(\boldsymbol{A} \odot \boldsymbol{M}_n)^T (\boldsymbol{A} \odot \boldsymbol{M}_n)\right]^{-1} (\boldsymbol{A} \odot \boldsymbol{M}_n)^T (\boldsymbol{w}_n \odot \boldsymbol{m}_n), \tag{13}$$

which is simplified by omitting repeated binary maskings,

$$\boldsymbol{b}_n = \left[\boldsymbol{A}^T (\boldsymbol{A} \odot \boldsymbol{M}_n)\right]^{-1} \boldsymbol{A}^T (\boldsymbol{w}_n \odot \boldsymbol{m}_n) . \tag{14}$$

The set of N estimates $\{\boldsymbol{b}_n, n = 1 \ldots N\}$ as in (14) generalizes the well known pseudoinverse LS solution in (9) to problems with missing data.

C-step – estimation of the column space Given $\boldsymbol{B}$, the estimate of each row $\boldsymbol{a}_m$ of the column space matrix $\boldsymbol{A}$ is obtained by proceeding in a similar way as in the R-step. We get, for each $m = 1 \ldots M$,

$$\boldsymbol{a}_m = (\boldsymbol{w}_m \odot \boldsymbol{m}_m) \boldsymbol{B}^T \left[(\boldsymbol{B} \odot \boldsymbol{M}_m) \boldsymbol{B}^T\right]^{-1}, \tag{15}$$

where in this case, for commodity, lowercase boldface letters denote rows rather than columns, and $\boldsymbol{M}_m = \mathbf{1}_{R \times 1} \boldsymbol{m}_m$.

5 Experimental Analysis

In this Section, we describe experiments that illustrate the behavior of the EM and CR algorithms and demonstrate their good performance.

2×2 matrices We start by a simple case that allows an illustrative graphical representation – estimating the 2×2 rank 1 matrix $\widetilde{\boldsymbol{W}}$ that best matches an observation $\boldsymbol{W}$ that misses one of its entries. In this case, the estimation error measured by the Frobenius norm in (3) and (8) can be expressed in terms of a single parameter θ. In fact, let the 2×2 rank 1 matrix $\widetilde{\boldsymbol{W}}$ be written in terms of its column and row spaces as in (7),

$$\widetilde{\boldsymbol{W}} = \boldsymbol{a}_{2\times 1} \boldsymbol{b}_{1\times 2} \in \mathcal{S}_1. \tag{16}$$

Without loss of generality, impose that the row vector $\boldsymbol{b}$ has unit norm and write it in terms of a row angle θ, $\boldsymbol{b} = [\cos\theta, \sin\theta]$. Now denote the minimum of (8) wrt the column space $\boldsymbol{a}$ for fixed row space $\boldsymbol{b}$, *i.e.*, for fixed θ, by $\boldsymbol{a}(\boldsymbol{W}, \theta)$, given by (15). The estimation error in (3) and (8) is then rewritten as a function of θ,

$$\text{error}(\theta) = \left\| \left(\boldsymbol{W} - \boldsymbol{a}(\boldsymbol{W}, \theta) \left[\cos\theta \, \sin\theta\right]\right) \odot \boldsymbol{M} \right\|_F . \tag{17}$$

Note that for any set of three entries of a 2×2 matrix, there is always a value for the forth entry that makes the rank of the matrix equal to one, *i.e.*, there is always a 2×2 rank 1 matrix $\widetilde{\boldsymbol{W}}$ that fits exactly the observed entries of $\boldsymbol{W}$. Thus, we have $\min \text{error}(\theta) = 0$.

The following examples illustrate the impact of the initialization on the behavior of the algorithms with experiments that use the same observations,

$$\widetilde{\boldsymbol{W}} = \begin{bmatrix} -1 & -1.95 \\ 2 & 3.9 \end{bmatrix} \in \mathcal{S}_1, \qquad \boldsymbol{M} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \boldsymbol{W} = \begin{bmatrix} -1 & -1.95 \\ 2 & ? \end{bmatrix}, \tag{18}$$

where "?" represents the unobserved entry w_{22} of the observation matrix $\boldsymbol{W}$.

Typical good behavior of EM and RC Using the initial estimate

$$\widehat{\boldsymbol{W}}_0 = \begin{bmatrix} -1 & -1.95 \\ 2 & 0 \end{bmatrix}, \tag{19}$$

we describe the evolution of the estimates $\widehat{\boldsymbol{W}}_k$ of the rank 1 matrix $\widetilde{\boldsymbol{W}}$ by plotting two equivalent representations of $\widehat{\boldsymbol{W}}_k$: **i)** its row space $\boldsymbol{b} = [b_1, b_2]$; and **ii)** the corresponding angle θ as defined above. The left plot of Fig. 1 shows the level curves of the error function as function of the row vector $\boldsymbol{b} = [b_1, b_2]$, superimposed with the evolution of the estimates of $\boldsymbol{b}$ for EM (dashed line) and RC (solid line). In this plot, the dotted line (optimum) are the row vectors that lead to zero estimation error. The right plot of Fig. 1 represents the same error function, now as a function of θ, as defined in (17), (dotted line) superimposed with the locations of the θ estimates for EM (dashed line) and RC (solid line).

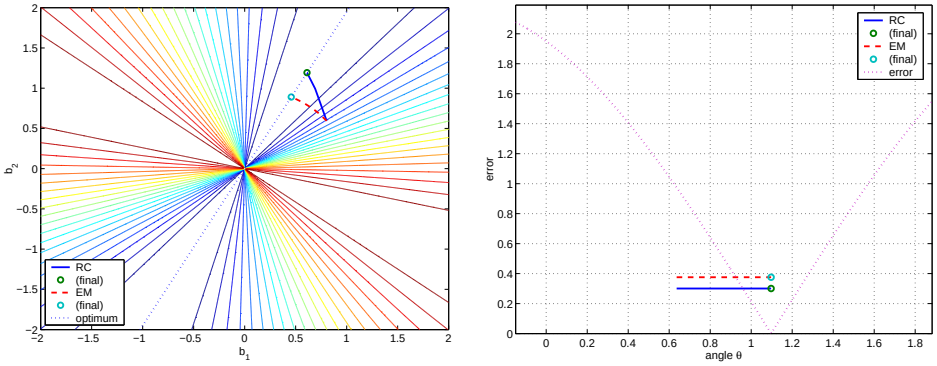

Fig. 1. Typical good behavior. Left: EM and RC trajectories. Right: error function.

From the left plot of Fig. 1, we see that both EM and RC trajectories start at the same point (due to the equal initialization) and converge to points in the optimal line. As expected, the EM estimates of the row space vector have constant unit norm (due to the normalization in the SVD) while the RC estimates don't. The good behavior of the algorithms is confirmed by the right plot of Fig. 1 that shows that both algorithms converge to a value of θ that makes error$(\theta) = 0$, *i.e.*, that minimizes error(θ).

To evaluate the convergence speed, we plot in Fig. 2 the evolution of the estimation error along the iterative process for both algorithms (in the left, in linear scale, in the right, in logarithmic scale). From Fig. 2, we see that RC converges in a smaller number of iterations than EM. Our experience with larger matrices in practical applications have confirmed the faster convergence of RC.

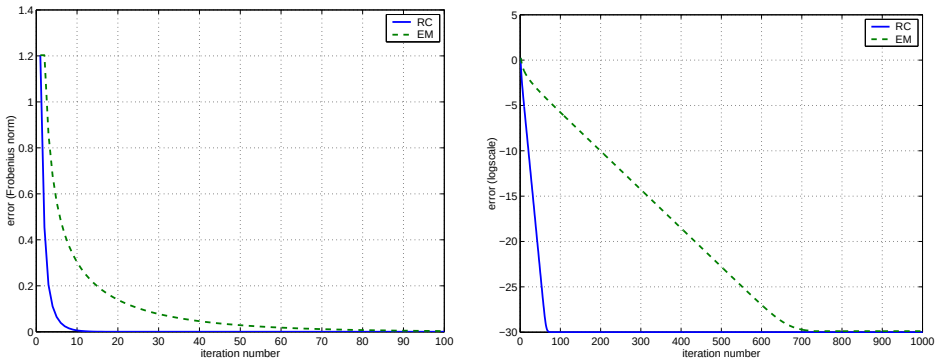

Fig. 2. Typical good behavior. Left: estimation error. Right: the same in logscale.

Large entries in rows or columns that miss data We now illustrate a drawback of the EM algorithm. Using the same data and the initial estimate

$$\widehat{\boldsymbol{W}}_0 = \begin{bmatrix} -1 & -1.95 \\ 2 & 22 \end{bmatrix}, \tag{20}$$

we get the first 1000 iterations of Fig. 3, which is as described above for Fig. 1.

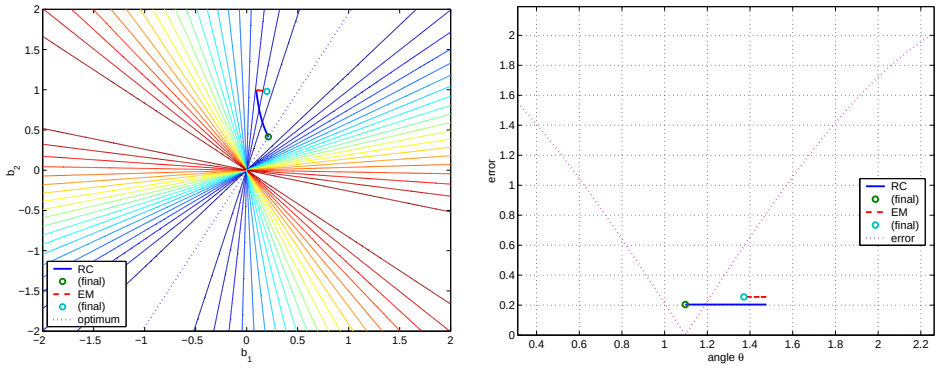

Fig. 3. Large initialization. Left: EM and RC trajectories. Right:error function.

From the left plot of Fig. 3, we see that, while RC converges to the optimal line, the estimates given by the EM almost doesn't change along the iterative process. This can also be seen in the right plot of Fig. 3, which shows that RC converges to the θ such that $\text{error}(\theta) = 0$, while EM, after 1000 iterations is still far from $\arg\min \text{error}(\theta)$. The left plot of Fig. 4 shows the evolution of the estimation errors. See that while the error of RC converges to zero in a few iterations, the error of EM almost doesn't decrease during the first 100 iterations.

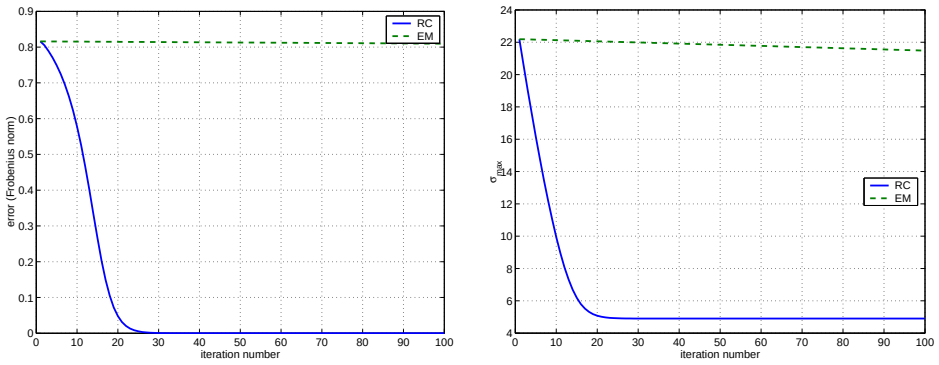

Fig. 4. Large initialization. Left: estimation error. Right: largest singular value of $\widehat{\boldsymbol{W}}_k$.

The bad behavior of EM is due to the large initial guess for w_{22} (note that $\widehat{w}_{0_{22}}$ is large when compared with the known entries of $\boldsymbol{W}$). Remember that EM starts by estimating the rank 1 matrix that best matches the initial guess $\widehat{\boldsymbol{W}}_0$. This rank 1 matrix is the matrix associated with the largest singular value of $\widehat{\boldsymbol{W}}_0$, which is highly constrained by the large spurious entry $\widehat{w}_{0_{22}} = 22$ (note that while the singular values of the solution matrix $\widetilde{\boldsymbol{W}}$ are $\sigma_1(\widetilde{\boldsymbol{W}}) \simeq 4.9$ and $\sigma_2(\widetilde{\boldsymbol{W}}) = 0$, the singular values of the initial guess $\widehat{\boldsymbol{W}}_0$ are $\sigma_1(\widehat{\boldsymbol{W}}_0) \simeq 22.2$ and $\sigma_2(\widehat{\boldsymbol{W}}_0) \simeq 0.8$ due to the large entry $\widehat{w}_{0_{22}} = 22$). Then, EM replaces the known entries of $\boldsymbol{W}$ in the new estimate (obtaining thus an estimate that is very close to the initial guess) and repeats the process. To better illustrate this very slow convergence of EM, we represent, in the right plot of Fig. 4, the evolution of the largest singular value of the estimate $\widehat{\boldsymbol{W}}_k$ for both algorithms. We see that while, as expected, the largest singular value of the RC estimates converges to the largest singular value of the solution $\widetilde{\boldsymbol{W}}$, $\sigma_1(\widetilde{\boldsymbol{W}}) \simeq 4.9$, the largest singular value of the first 100 iterations of the EM estimates changes very slowly from its initial value $\sigma_1(\widehat{\boldsymbol{W}}_0) \simeq 22.2$.

The behavior just described also happens in situations other than the initial guesses of the unknown entries being too large when compared to the other entries. In fact, we observed the same behavior in situations where the observation matrix had large entries in rows or columns that contained missing entries. In these situations, due to those large entries, even small values for initial guess of the unknown entries had large impact on the row and column singular vectors associated with the large singular values that determined the best rank deficient approximations to the complete data matrices involved in EM. This lead to the same kind of very slow convergence illustrated in Fig. 3 and Fig. 4.

Large matrices with random initialization We tested the algorithms with noisy partial observations of rank deficient matrices. We used matrices with dimensions ranging from 2×2 to 200×200 and rank from 1 to 6. The percentage

of missing entries ranged from 10% to 80%. We initialized both EM and RC with random values for the unknown entries.

To illustrate the influence of the initialization on the convergence of the algorithms, Fig. 5 shows the percentage of experiments that converged (to an estimate close enough to the ground truth) in less than 100 iterations, as a function of the mean value of the random guesses for the unknown entries. Although representative for the entire range of experiments done, the results in the plots of Fig. 5 were obtained with noisy observations of 24×24 rank 4 matrices that missed 70% of their entries. The left plot of Fig. 5 shows three lines, each obtained by using EM with data generated with a ground truth matrix whose elements had mean 0.001, 1, and 1000. The percentages of convergence for the RC algorithm are in the right plot.

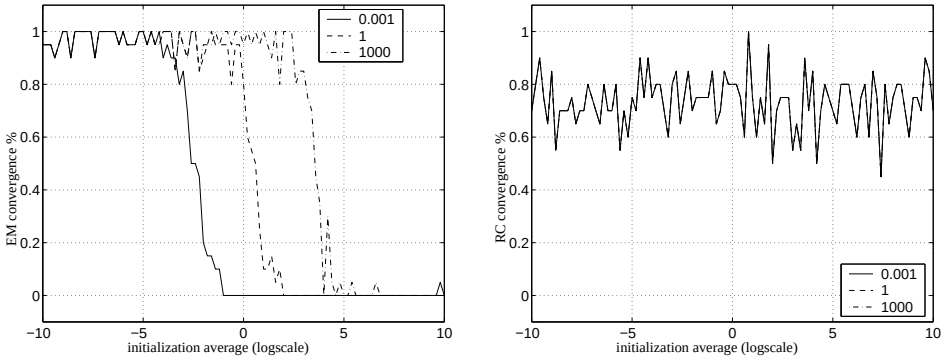

Fig. 5. Percentage of convergent experiments as a function of the mean value of the initial random guesses for the unknown entries. Left: results for the EM algorithm with matrices whose entries have mean 10^{-3}, 1, and 10^{3}. Right: results for the RC algorithm.

The left plot of Fig. 5 shows that the EM algorithm converges almost always if the mean value of the initial guesses for the missing entries is smaller than the mean value of the observed entries. When we increase the values of the initial estimates of the missing entries, the percentage of convergence decreases abruptly, becoming close to zero when those values become much larger than the ones of the observed entries. This is in agreement with the behavior illustrated in the example of Fig. 3. In opposition, the right plot of Fig. 5 shows that the behavior of the RC algorithm is somewhat independent of the order of magnitude of the initialization. The experiments that lead to a non-convergent behavior of RC were such that the matrices whose inverse is computed in (15) and (14) were close to singular. We thus conclude that it is very important in practical applications to provide good initial estimates for both EM and RC algorithms.

Finally, we note that the relevance of a good initialization goes behind avoiding non-convergent behavior. In fact, we observed that both the amount of miss-

ing data and the noise level have strong impact on the algorithm's convergence speed. Thus, when dealing with large percentages of missing entries and high levels of noise, as it may arise in practice, a better initialization not only improves the chance of a convergent behavior but also leads to a faster convergence.

Heuristic initialization We now use an initial guess $\widehat{\boldsymbol{W}}_0$ obtained by combining the column and row spaces given by the SVDs of the known submatrices of $\boldsymbol{W}$ [24]. In our tests, with this simple initialization procedure, 100% of the runs of EM and RC converged to the ground truth matrix in a very small number of iterations, typically less than 10, even for high levels of noise.

The plot of Fig. 6 represents the average entry estimation error, defined as $\overline{error}(\widehat{\boldsymbol{W}}) = \|(\widehat{\boldsymbol{W}} - \widetilde{\boldsymbol{W}}) \odot \boldsymbol{M}\|_F / \sqrt{\sum_{i,j} m_{ij}}$, where $\sum_{i,j} m_{ij}$ accounts for the number of observed entries, after 20 iterations of EM and RC algorithms, for noisy observations of a 24×24 rank 4 matrix $\widetilde{\boldsymbol{W}}$, with 70% missing data, as a function of noise standard deviation. We see that the average entry estimation errors after 20 iterations are below 10^{-8} for noise standard deviation ranging from $10^{-2.5}$ to $10^{2.5}$ (the mean value of the entries of the ground truth matrix $\widetilde{\boldsymbol{W}}$ is 1). This shows that, with a simple initialization procedure, both EM and RC algorithms converge to the optimal solution, even for very noisy observations. Furthermore, we conclude that the main impact of the observation noise is on the EM and RC convergence speeds – the slightly higher average error values on the right region of the plot of Fig. 6 indicates that the estimates were still converging to the optimal solution after 20 iterations.

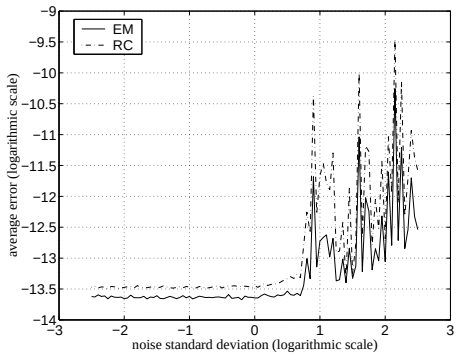

Fig. 6. Estimation errors for EM and RC as functions of the noise standard deviation.

Computational cost As referred above, the EM and RC algorithms converge in a very small number of iterations when initialized by the heuristic procedure in [24]. We now report an experimental evaluation of the computational costs of each iteration of EM and RC as functions of the observation matrix dimension.

We used $N \times 24$ observation matrices with missing data corresponding to a $(N-4) \times 20$ submatrix. The plots in Fig. 7 represent the number of MatLab©

floating point operations (FLOPS) and the computation time per iteration, as functions of N. From the left plot, we see that the number of FLOPS per iteration of the EM algorithm is larger than one of the RC algorithm. Furthermore, the FLOPS count for EM increases exponentially with N (due to the SVD computation) while for RC it increases linearly with N. Thus, although the computation times in the right plot of Fig. 7 are smaller for EM than for RC (the reason being the very efficient MatLab© implementation of the SVD), we conclude that RC is computationally simpler than EM. RC is even as simple as the methods to deal with complete matrices, since the most efficient way to compute the SVD is to use the power method [18] of which RC is a simple generalization.

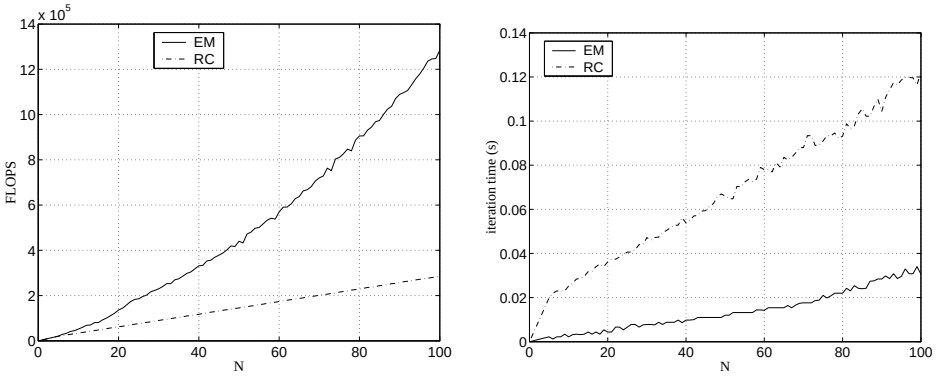

Fig. 7. Computational cost of each iteration of EM and RC. Left: number of MatLab© floating point operations (FLOPS). Right: computation time.

6 Application: 3D Structure from Video Sequences

We illustrate the application of the matrix estimation algorithms to the problem of recovering 3D rigid shape and 3D motion from video. As with the majority of current approaches, we infer the 3D positions of a set of features points and the 3D motion of the camera by first determining the 2D trajectories of the feature points on the image plane. Among the approaches to compute the 3D rigid structure parameters from the 2D projections, we use the *factorization method* of Tomasi and Kanade [8], now popular due to its robustness and simplicity. In this method, the observed trajectories are collected in an observation matrix $\boldsymbol{W}$ that is rank deficient in a noiseless situation. The parameters describing the 3D shape and 3D motion are estimated from the factors of the rank deficient matrix $\widetilde{\boldsymbol{W}}$ that best matches $\boldsymbol{W}$. In [8], the authors use the SVD to compute $\widetilde{\boldsymbol{W}}$ when $\boldsymbol{W}$ is completely observed. In practice, however, due to the limited field of view and the occlusion, the observation matrix $\boldsymbol{W}$ may miss several of its entries. We use our algorithms to compute $\widetilde{\boldsymbol{W}}$ when $\boldsymbol{W}$ has missing data. This way, we take into

account the rigidity of the scene along the entire video sequence, leading to a more constrained problem (and a more accurate solution) than the one obtained by processing independently several small subsets of frames. We describe below experiments with synthesized and real video sequences.

Synthetic data – cylinder We synthesized noisy versions of the 2D trajectories of 372 feature points located on the 3D surface of a rotating cylinder. Then, we simulated occlusion and inclusion by removing significant segments of those trajectories. The left plot of Fig. 8 shows one of the 50 synthesized frames. The small circles denote the noiseless projections and the points denote their noisy version, *i.e.*, the data that is observed. Note that only an incomplete view portion of the cylinder is observed in each frame.

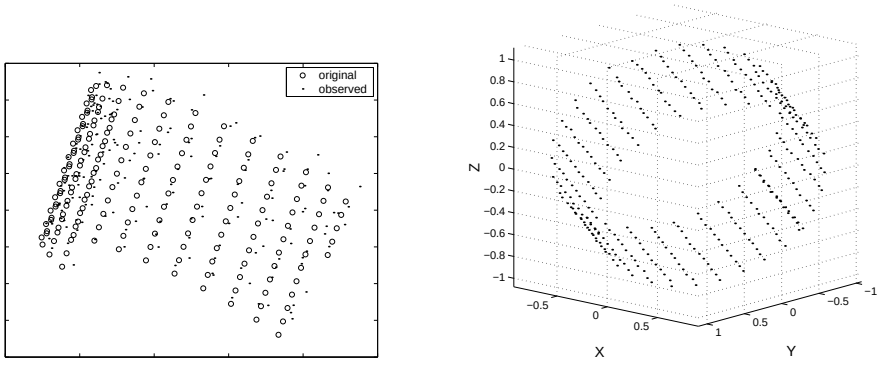

Fig. 8. Cylinder sequence. Left: one synthetic frame. Right: estimates of the 3D positions of the features points.

The data from the cylinder sequence was then collected on a 100×372 observation matrix $\boldsymbol{W}$ with 9537 unknown entries ($\simeq 26\%$ of the total number). We used the RC algorithm to estimate the rank deficient matrix $\widetilde{\boldsymbol{W}}$ that best matched $\boldsymbol{W}$ and computed the 3D structure by using the factorization method [8]. The right plot of Fig. 8 plots the final estimate of the 3D shape. We see that the complete cylinder is recovered. Due to the incorporation of the rigidity constraint, the 3D positions of the features points are accurately estimate even in the presence of very noisy observations (compare the plots in Fig. 8).

Real video – ping-pong ball We used a real-life video clip available at [26]. This clip shows a rotating ping-pong ball with painted dots. The left image of Fig. 9 shows the first of the 52 video frames of the ball sequence. We used simple correlation techniques to track a set of 64 feature points. Due to the camera-ball 3D rotation, the region of the ball that is visible changes along time, leading to an observation matrix with $\simeq 41\%$ missing entries. By proceeding as described for the previous experiment, we estimated the 3D shape shown on the right image of Fig. 9, which shows that our method succeeded in recovering the spherical surface of the ball.

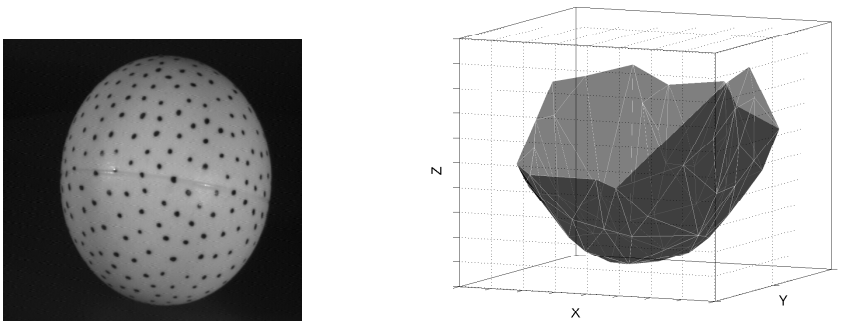

Fig. 9. Ping-pong ball video clip. Left: first frame. Right: estimated 3D shape.

Real video – Rubik's cube This video clip – see two representative frames on the left and middle images of Fig. 9 – shows a Rubik's cube rotating around a vertical axis. In the leftmost image of Fig. 9, we superimposed with the video frame the visible features and the initial parts of their trajectories. Due to the occlusion, feature points enter and leave the scene. To emphasize the advantages of using the algorithms described in this paper, we first applied to a segment of the Rubik's cube video clip the factorization method of Tomasi and Kanade [8] for complete data, obtaining the 3D shape represented on the left side of Fig. 11. This model was obtained with 28 features and 18 frames.

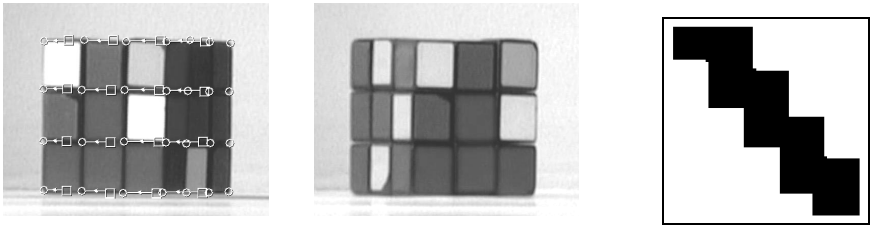

Fig. 10. Rubik's cube video clip. Left: frame with visible features and corresponding partial trajectories. Middle: another frame. Right: binary mask matrix $\boldsymbol{M}$ representing the incomplete data – black regions correspond to entries $m_{ij} = 1$ meaning that w_{ij} is observed, *i.e.*, feature j is visible in frame i; white regions represent the opposite.

We then collected the entire set of the visible parts of the trajectories of 64 features across 85 frames in a 170×64 incomplete observation matrix $\boldsymbol{W}$. The structure of the missing part of $\boldsymbol{W}$ is coded by the 170×64 binary mask $\boldsymbol{M}$ represented on the right side of Figure 10. The number of missing entries in $\boldsymbol{W}$ was about 62%. Since the 3D motion in this video clip is a 3D rotation, the observation matrix would be rank 3 in a noiseless situation [8]. We used the RC algorithm to estimate the rank 3 matrix that best matches the incomplete data in $\boldsymbol{W}$ and then the factorization method to compute the 3D shape shown on

the right image of Fig. 11. This simple example illustrates how RC algorithm trivializes the usually hard task of merging partial estimates of a 3D model. In the right image of Fig. 11, the top face is missing because the position the cube model is shown in that image was not seen in the original video clip.

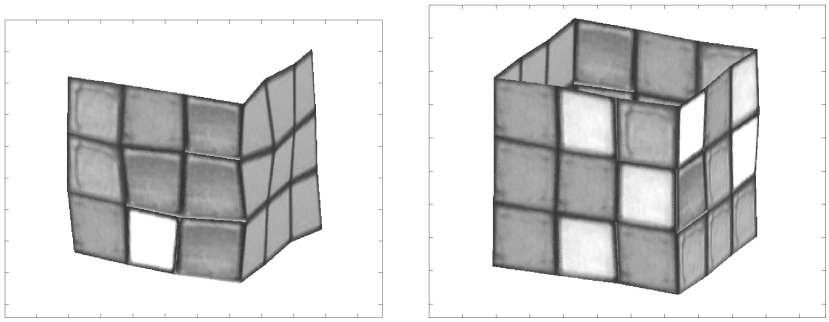

Fig. 11. Texture mapped 3D shape recovered from the Rubik's cube video clip. Left: incomplete model obtained by using the factorization method of Tomasi and Kanade [1]. Right: complete shape recovered by our method – factorization with missing data.

The advantage of using RC or EM to recover 3D rigid structure is two-fold. First, while recovering a complete 3D model by fusing partial models as the one on the left side of Fig. 11 is a complex task, our method recovers directly the complete model shown on the right side of Fig. 11. Second, rather than processing subsets of the set of features and frames at disjoint steps, our method uses all the information available in a global way, leading to more accurate 3D shapes as illustrated by the 3D models in Fig. 11.

7 Conclusion

We developed two iterative algorithms, *Expectation-Maximization* (EM) and *Row-Column* (RC), that estimate rank deficient matrices from partial observations. Our experiments showed that both algorithms converged to the correct estimate whenever initialized by a simple procedure and that RC algorithm is computationally cheaper and more robust than EM. We used RC to recover 3D structure from video sequences with self-occluding objects.

References

1. Basri, R., Ullman, S.: The alignment of objects with smooth surfaces. Computer Graphics, Vision, and Image Processing: Image Understanding **57** (1988)
2. Ullman, S., Basri, R.: Recognition by linear combinations of models. IEEE TPAMI **13** (1991)

3. Shashua, A.: Geometry and photometry in 3D visual recognition. MIT TR 1401, Massachussets Institute of Technology, MA, USA (1992)
4. Moses, Y.: Face recognition: generalization to novel images. PhD thesis, Weizmann Institute of Science (1993)
5. Belhumeur, P., Kriegman, D.: What is the set of images of an object under all possible lighting conditions ? In: IEEE CVPR. (1996)
6. Zelnik-Manor, L., Irani, M.: Multi-frame alignement of planes. In: IEEE CVPR, Fort Collins CO, USA (1999)
7. Zelnik-Manor, L., Irani, M.: Multi-view subspace constraints on homographies. In: IEEE ICCV, Kerkyra, Greece (1999)
8. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9** (1992)
9. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Proc. of ECCV, Cambridge, UK (1996)
10. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. IEEE TPAMI **19** (1997)
11. Morita, T., Kanade, T.: A sequential factorization method for recovering shape and motion from image streams. IEEE TPAMI **19** (1997)
12. Aguiar, P.M.Q., Moura, J.M.F.: Factorization as a rank 1 problem. In: IEEE CVPR, Fort Collins CO, USA (1999)
13. Irani, M., Anandan, P.: Factorization with uncertainty. In: Proc. of ECCV, Dublin, Ireland (2000)
14. Shapiro, L.: Affine Analysis of Image Sequences. Cambridge University Press, Cambridge, UK (1995)
15. Quan, L., Kanade, T.: A factorization method for affine structure from line correspondences. In: IEEE CVPR, San Francisco CA, USA (1996)
16. Morris, D., Kanade, T.: A unified factorization algorithm for points, line segments and planes with uncertainty models. In: IEEE ICCV. (1998)
17. Aguiar, P.M.Q., Moura, J.M.F.: Three-dimensional modeling from two-dimensional video. IEEE Transactions on Image Processing **10** (2001)
18. Golub, G.H., Van-Loan, C.F.: Matrix Computations. The Johns Hopkins University Press (1996)
19. Jacobs, D.: Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In: IEEE CVPR, St Barbara, USA (1997)
20. Maruyama, M., Kurumi, S.: Bidirectional optimization for reconstructing 3D shape from an image sequence with missing data. In: IEEE ICIP, Kobe, Japan (1999)
21. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. John Wiley & Sons, New York (1997)
22. Wiberg, T.: Computation of principal components when data are missing. In: Proc. of the Second Symposium Computational Statistics, Berlin, Germany (1976)
23. Shum, H., Ikeuchi, K., Reddy, R.: Principal component analysis with missing data and its applications to polyhedral object modeling. IEEE TPAMI **17** (1995)
24. Guerreiro, R.F.C., Aguiar, P.M.Q.: 3D structure from video streams with partially overlapping images. In: IEEE ICIP, New York, USA (2002)
25. http: //www.isr.ist.utl.pt/~aguiar/code.html (2002)
26. http: //www-2.cs.cmu.edu/~cil/vision.html (2002)