Factorization with Missing Data for 3D Structure Recovery

Rui F. C. Guerreiro and Pedro M. Q. Aguiar
Institute for Systems and Robotics / Instituto Superior Técnico, Lisboa, Portugal
Email: {rfcg,aguiar}@isr.ist.utl.pt

Abstract—**Matrix factorization methods are now widely used to recover 3D structure from 2D projections [1]. In practice, the observation matrix to be factored out has missing data, due to the limited field of view and the occlusion that occur in real video sequences. In opposition to the optimality of the SVD to factor out matrices without missing entries, the optimal solution for the missing data case is not known. In reference [2] we introduced suboptimal algorithms that proved to be more efficient than previous approaches to the factorization of matrices with missing data. In this paper we make an experimental analysis of the algorithms of [2] and demonstrate their performance in virtual reality and video compression applications. We conclude that these algorithms are: i) adequate to the amount of missing entries that may occur when processing real videos; ii) robust to the typical level of noise in practical applications; and iii) computationally as simple as the factorization of matrices without missing entries.**

I. Introduction

Computing 3D structure from video has applications in fields ranging from virtual reality and digital video to robotics. Virtual reality applications often require 3D models of real world objects. The manual description of the 3D models or the highly expensive laser systems can be avoided by using methods that construct the 3D models in an automatic way from ordinary video. In digital video, the automatic recovery of 3D structre enables very efficient model-based coding techniques. In robotics, the video camera is an increasingly popular sensor for autonomous vehicles that need to construct a 3D model of the environment for navigation and recognition purposes. Since the most powerful cue to infer 3D structure from video is the 2D motion of the brightness pattern induced on the image plane, the recovery of 3D structure from video is usually known as the *structure from motion* (SFM) problem. Early approaches to SFM used a single pair of frames and were shown to be very sensitive to image noise. The key to the robustness of the SFM methods was on exploiting the rigidity of the scene across a larger set of frames. Unfortunately, although using multiple frames leads to a more constrained problem, the multi-frame formulation is also more complex – the number of unknowns grows due to the larger number of camera poses to estimate.

Among the approaches to multi-frame SFM, the factorization method, introduced in the early nineties [1], has become popular. It captures rigidity in an algebraic way. In [1], the trajectories of the 2D projections of a set of feature points are collected in an observation matrix. Due to the rigidity of the scene, the observation matrix is rank 4 in a noiseless situation. The 3D rigid shape and the 3D motion are computed from the factors of the rank 4 matrix that best matches the observation matrix. The work of [1] was later extended in several directions, e.g., geometric projection model [3], 3D shape description [4], and recursive formulation [5]. The factorization methods [1], [3]-[5] are limited to use feature points whose projections are visible through the entire video sequence, so that the best rank 4 approximation of the observation matrix, which is completely known, is obtained from its SVD. This is a serious limitation since, in real life video clips, due to scene self-occlusion and limited field of view, the projections of the points of interest are not always visible. In practice, it is then necessary to fuse a set of partial factorization estimates to obtain the complete solution to 3D structure. This isn't a simple task and leads in general to inaccurate solutions. In opposition to this local formulation, the rigidity of the scene is captured in a global way when the observation matrix is allowed to contain missing data. The problem becomes then how to find the best rank 4 approximation of an observation matrix that has missing entries.

There is no equivalent to the SVD for matrices with missing entries. Very few attempts have been made to extend the factorization method to the missing data case. Suboptimal methods can be found in [6] and [7]. In [2], we develop two iterative algorithms that converge to the optimal factorization of a matrix with missing entries, when properly initialized. Reference [2] also describes an initialization procedure. The source code for the algorithms of [2] is available from the WWW link [8]. In this paper we make an experimental performance analysis of the algorithms of [2] and outline applications of the factorization algorithms to virtual reality and digital video compression tasks. We tested the two iterative algorithms in what respects to the computational cost and to the impact of the: i) initialization, ii) noise, and iii) missing entries. We also studied the behavior of the initialization algorithm. Our conclusions are that the algorithms of [2] are adequate to real video processing tasks. Our experience showed that, for the typical noise and missing data, the initial estimate, provided by the initialization algorithm, enables both iterative algorithms to converge: i) to the global optimum; and ii) in a very small number of iterations.

Paper organization In section II we overview the algorithms for factoring out matrices with missing data. Section III describes the experimental analysis of the algorithms. In section IV we describe virtual reality and video compression applications. Section V concludes the paper.

Work partially supported by FCT project POSI/SRI/41561/2001.

0-7803-7713-3/02/$17.00 © 2002 IEEE

II. OVERVIEW OF THE ALGORITHMS

In reference [2], we propose two iterative algorithms to compute the rank 4 matrix $\widetilde{W}_{M\times N}$ that best matches the observation matrix $W_{M\times N}$ that has missing entries. The first iterative algorithm is based on an *Expectation-Maximization* procedure that has been successfully used in several signal processing tasks involving missing data [9]. The second algorithm is a generalization of the *power method* that is widely used to compute SVD-based rank deficient approximations of matrices without missing entries [10]. As for any iterative algorithm, initialization is a relevant issue. In [2], the initial estimate of the rank 4 matrix $\widetilde{W}$ is obtained by combining the column and row spaces of the known portions of W.

Expectation-Maximization (EM) The EM algorithm estimates in alternate steps: i) the missing entries of W – E-step; and ii) the rank 4 matrix $\widetilde{W}$ that best matches the "completed version" of matrix W – M-step. The solution for the E-Step is simply given by the corresponding entries of the previous estimate of $\widetilde{W}$. The M-step is solved by using the SVD [2].

Two-step (TS) iterative algorithm The TS algorithm computes, alternately, two matrices $A_{M\times 4}$ and $B_{4\times N}$ whose product is the solution matrix $\widetilde{W} = AB$. In step i), we assume the column space matrix B is known and estimate the row space matrix A; in step ii), we estimate B for known A. Both steps have closed form solution [2].

III. EXPERIMENTAL ANALYSIS

We tested the EM and TS algorithms by synthesizing observation matrices W likely to occur when processing real videos. In particular, we studied the behavior of the algorithms in terms of the observation noise power and the amount of missing data. Since EM and TS are iterative algorithms that need an initial estimate of the rank 4 matrix $\widetilde{W}$, our experiences addressed in first place the impact of the initialization.

Influence of the initialization We generated several ground truth rank 4 matrices W_t. The algorithms receive as input an incomplete observation of W_t determined by the binary mask M, i.e., $m_{ij} = 1$ if w_{ij} is known and $m_{ij} = 0$ otherwise. We measure the estimation error as the mean over the known entries,

$$\bar{e} = \frac{\left\|\left(\widetilde{W} - W_t\right) \odot M\right\|_F}{\sqrt{\sharp M}}, \qquad (1)$$

where $\widetilde{W}$ is the estimated rank 4 matrix; $\odot$ stands for the elementwise product, also known as the Hadamard product; $\|.\|_F$ represents the Frobenius norm; and $\sharp M = \sum_{i,j} m_{ij}$, i.e., $\sharp M$ is the number of the number of known entries of W.

With ground truth matrices W_t with dimensions ranging from 4×4 to 200×100 and missing data from 0% to 80%, we run the initialization algorithm and, subsequently, both the EM and TS iterative refinements. In all these experiments, the errors (1) obtained after convergence were of the same magnitude of the machine precision. Both algorithms converged in a few iterations, typically less than 10. Thus, when adequately initialized, both EM and TS algorithms converged always to the optimal solution in a small number of iterations.

To evaluate the impact of the good initial estimate, we run EM and TS with random initializations. We observed that the behavior of the iterative algorithms is not easily predicted – they may take an huge number of iterations to converge to the global optimum or even diverge. To illustrate this fact, we varied the mean value of the initial estimates of $\widetilde{W}$. The plots in Figure 1 show the estimated convergence probabilities as functions of the mean value of the initial estimate of $\widetilde{W}$ for three 24×24 ground truth matrices W_t with mean value 0.001, 1, and 1000 and 70% missing entries. The left plot of Figure 1 shows that the EM algorithm converges to the global optimum when the mean value of the initial estimate $\widetilde{W}$ is smaller than the mean value of W_t and diverges or converges to a wrong solution if the opposite. This fact is explained by the relative impact on the SVD (in the M-step of the EM algorithm, see section II) of the entries corresponding to the random initial estimate and the entries corresponding to the observed data. From the right plot of Figure 1, we see that the probability of convergence for the TS algorithm is roughly independent of the mean value of the initialization – in these experiments, the TS algorithm converges to the global optimum about 75% of the runs.

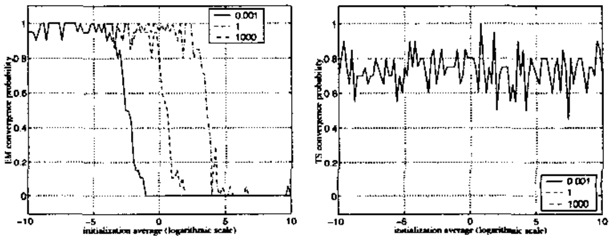

Fig. 1. Probability of convergence with random initialization. Left: EM algorithm. Right: TS algorithm.

We thus conclude that the initialization procedure of [2] has a relevant impact on the good convergence of both EM and TS algorithms. In fact, the initialization algorithm provides an initial estimate that is close enough to the optimal solution to guarantee: i) good convergence – in our tests, 100% of the runs lead to the global optimum, and ii) fast convergence – the algorithms stop in a very small number of iterations, typically less than 10.

Sensitivity to the noise In real-life applications, the observation matrix W is contaminated with noise due to feature tracking errors. In this experimental performance analysis we used white Gaussian noise. The observation matrix W is modeled as a noisy version on the ground truth W_t, $W = W_t + N$, where the additive noise N is zero mean.

We tested our algorithms with noisy observations of matrices with dimensions from 4×4 to 200×100 and missing data from 0% to 80%. As a representative example, the plots in Figure 2 represent the average estimation error given by (1) as a function of the noise variance for a 24×24 observation matrix $\boldsymbol{W}$ with 70% missing entries. The error of the initial estimate is represented on the left side plot. The estimation error obtained after 20 iterations of either EM or TS is represented on the right.

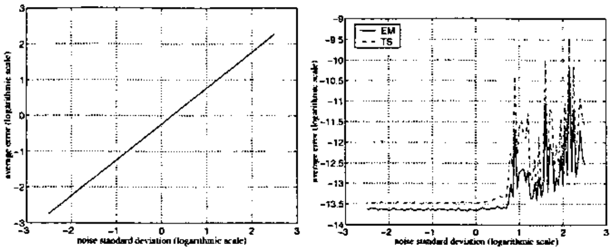

Fig. 2. Sensitivity to the noise. Left: initial estimate. Right: EM and TS final estimates.

The left plot of Figure 2 shows that the average error of the initial estimate increases linearly with the noise standard deviation. The right plot of Figure 2 shows that the average estimation errors after 20 iterations of both EM and TS algorithms is below 10^{-8} for noise standard deviation ranging from $10^{-2.5}$ to $10^{2.5}$ (the mean value of the ground truth matrix $\boldsymbol{W}_t$ is 1). This shows that EM and TS converge to the optimal solution even in the presence of high levels of feature tracking errors. In fact, we concluded that the main impact of the observation noise is on the EM and TS convergence speeds – the slightly higher average error values on the right region of the right plot of Figure 2 indicate that the process was still converging to the optimal solution after the 20 iterations.

Sensitivity to the missing data A relevant issue is the robustness of the factorization algorithms to the missing data. Our experience showed that the structure of the binary mask matrix M representing the known data is by far more important than the overall percentage of missing entries of $\boldsymbol{W}$. When recovering 3D structure from video, feature points enter and leave the scene in a continuous way, thus the typical structure of M is as represented in the right plot of Figure 5. We tested the factorization algorithms with several mask matrices M with this typical structure. In all these experiments, the algorithms converged always to the optimal solution in a very small number of iterations, independently of the percentage of missing data.

As for the impact of the noise discussed above, the percentage of missing entries has impact on the convergence speed. To demonstrate this fact, we run the iterative algorithms for 24×24 observation matrices $\boldsymbol{W}$ with missing entries corresponding to a $N \times 20$ submatrix. To better illustrate the dependence of the algorithms behaviors on the percentage of missing data, i.e., on N, we used in this experiment random initializations. The plot in Figure 3 represents the average errors after 20 iterations of EM and TS as functions of N. This plot shows that the larger is the portion of the matrix that is observed, i.e., the smaller is the percentage of missing data, the lower is the estimation error after 20 iterations, i.e., the faster is the convergence of the iterative algorithms.

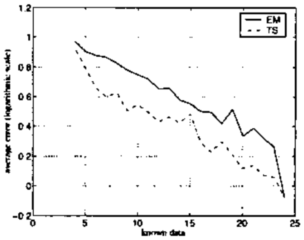

Fig. 3. Sensitivity to the missing data. Errors after 20 iterations of the EM and TS algorithms with random initialization.

Computational cost As referred above, the EM and TS iterative algorithms converge in a very small number of iterations when initialized by the initialization procedure of [2]. In this subsection, we report on the computational costs of each iteration of EM and TS as functions of the dimension of the observation matrix.

We used $N \times 24$ observation matrices $\boldsymbol{W}$ with missing data corresponding to a $(N-4) \times 20$ submatrix. The plots in Figure 4 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 TS algorithm. Furthermore, the FLOPS count for EM increases exponentially with N (due to the SVD computation) while for TS it increases linearly with N. Thus, although the computation times in the right plot of Figure 4 are smaller for EM than for TS (the reason being the very efficient MatLab© implementation of the SVD – different FLOPS have different computation times that depend on the hardware and software), we conclude that TS is computationally much simpler than EM. TS is even as simple as the methods to factorize complete matrices, since the most efficient way to compute the SVD is to use the power method [10] of which TS is a simple generalization,

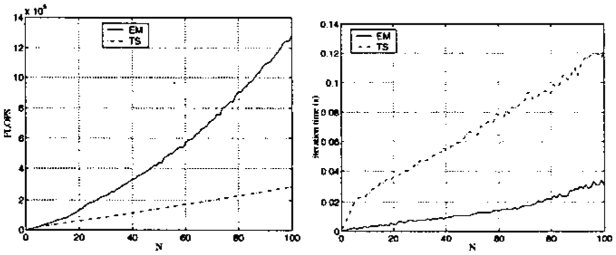

Fig. 4. Computational cost of each iteration of EM and TS. Left: number of FLOPS. Right: computation time.

IV. APPLICATIONS

To illustrate virtual reality and video compression applications of our algorithms, we used the Rubik's cube video clip. This clip – see a representative frame on the left image of Figure 5 – shows a Rubik's cube rotating around a vertical axis.

3D modeling for virtual reality We used simple correlation techniques to track feature points across the Rubik's cube video clip. In the left side of Figure 5, 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 our factorization with missing data, we first applied to a segment of the Rubik's cube video clip the factorization method of Tomasi and Kanade [1] for complete data, obtaining the 3D shape represented on the left side of Figure 6. This model was obtained with 28 features and 18 frames.

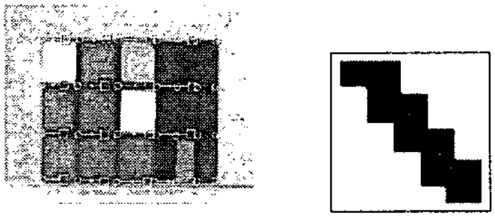

Fig. 5. Rubik's cube video clip. Left: first frame with visible features and corresponding partial trajectories. Right: binary mask matrix M representing the incomplete data – black regions correspond to entries $m_{ij} = 1$ meaning that w_{ij} is observed, i.e., feature j is visible in frame i; white regions represent the opposite.

We then applied our method. We collected the entire set of the visible parts of the trajectories of 64 features across 85 frames in a 170 × 64 incomplete observation matrix W. The structure of the missing part of $\boldsymbol{W}$ is coded by the 170 × 64 binary mask M represented on the right side of Figure 5. The number of missing entries in $\boldsymbol{W}$ is about 62% of the total number of entries. We applied our factorization algorithm to the incomplete observation matrix $\boldsymbol{W}$ and recovered the 3D model represented on the right side of Figure 6. The top face is missing because the position the cube model is shown was not seen in the original video clip.

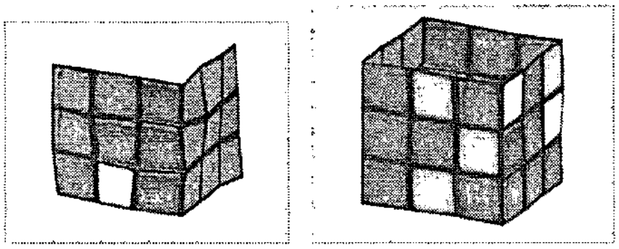

Fig. 6. Texture mapped 3D shape recovered from the Rubik's cube video clip. Left: incomplete model obtained by using the factorization method of Tomasi and Kanade [1] for complete data. Right: complete shape recovered by our method – factorization with missing data.

The advantage of our method is two-fold. First, while recovering a complete 3D model by fusing partial models as the one on the left side of Figure 6 is a complex task, our method recovers directly the complete model shown on the right side of Figure 6. Second, rather than processing subsets of the sets 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 Figure 6.

Video compression The 3D models recovered by factorization with missing data can be used to represent in an efficient way the original video sequence as proposed in [11] – the video sequence is represented by the 3D shape, texture, and 3D motion of the objects. This leads to significant bandwidth saving since once the 3D shape and texture of the objects have been transmitted, their 3D motion is transmitted with a few bytes per frame.

We used this methodology to compress the entire sequence of 2161 frames of the Rubik's cube video clip. The compression ratio (relative to the original JPEG compressed frames) was approximately 10^3. Figure 7 shows sample original frames (top row) and the corresponding compressed frames (bottom row). The differences of lighting between the top and bottom images are due to the constancy of the texture of the 3D model.

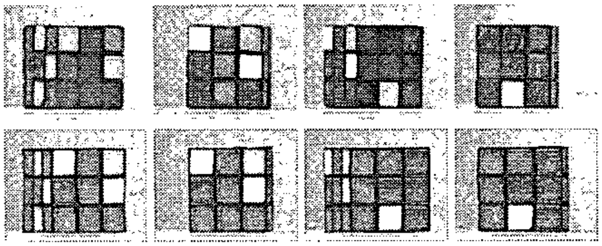

Fig. 7. 3D model-based digital video compression example. Top row: original frames. Bottom row: compressed frames. Compression ratio approx. 10^3.

V. CONCLUSION

We presented an experimental analysis of the algorithms EM and TS that factor out matrices with missing entries. Our analysis shows that EM and TS are well suited to the recovery of 3D rigid structure from video sequences. We demonstrate the performance of the factorization algorithms in virtual reality and video compression applications.

REFERENCES

[1] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), 1992.

[2] R. F. C. Guerreiro and P. M. Q. Aguiar. 3D structure from video streams with partially overlapping images. To appear in *IEEE ICIP*, New York, USA, September 2002.

[3] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. on PAMI*, 19(3), 1997.

[4] P. M. Q. Aguiar and J. M. F. Moura. Three-dimensional modeling from two-dimensional video. *IEEE Trans. on Image Processing*, 10(10), 2001.

[5] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Trans. on PAMI*, 19(8):858–867, 1997.

[6] D. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *IEEE CVPR*, Santa Barbara CA, USA, 1997.

[7] M. Maruyama and S. Kurumi. Bidirectional optimization for reconstructing 3D shape from an image sequence with missing data. In *IEEE ICIP*, Kobe, Japan, 1999.

[8] `http://www.isr.ist.utl.pt/~aguiar/code.html`.

[9] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, 1997.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[11] P. M. Q. Aguiar and J. M. F. Moura. Fast 3D modelling from video. In *IEEE MMSP*, Copenhagen, Denmark, 1999.