

Incorporating Constraints and Prior Knowledge into Factorization Algorithms - an Application to 3D Recovery

Amit Gruber and Yair Weiss

School of Computer Science and Engineering,
The Hebrew University of Jerusalem,
Jerusalem, Israel 91904
{amitg,yweiss}@cs.huji.ac.il

Abstract. Matrix factorization is a fundamental building block in many computer vision and machine learning algorithms. In this work we focus on the problem of “structure from motion” in which one wishes to recover the camera motion and the 3D coordinates of certain points given their 2D locations. This problem may be reduced to a low rank factorization problem. When all the 2D locations are known, singular value decomposition yields a least squares factorization of the measurements matrix. In realistic scenarios this assumption does not hold: some of the data is missing, the measurements have correlated noise, and the scene may contain multiple objects. Under these conditions, most existing factorization algorithms fail while human perception is relatively unchanged. In this work we present an EM algorithm for matrix factorization that takes advantage of prior information and imposes strict constraints on the resulting matrix factors. We present results on challenging sequences.

1 Introduction

The problem of “structure from motion” (SFM) has been studied extensively [15, 14, 11, 10, 2] in computer vision: Given the 2D locations of points along an image sequence, the goal is to retrieve the 3D locations of the points. Under simplified camera models, this problem reduces to the problem of matrix factorization [15].

Using SVD, the correct 3D structure can be recovered even if the measurements matrix is contaminated with significant amounts of noise and if the number of frames is small [15].

However, in realistic situations the measurement matrix will have *missing* entries, due to occlusions or due to inaccuracies of the tracking algorithm. A number of algorithms for factorization with missing data [15, 10, 14, 2] have been suggested. While some of these algorithms obtain good results when the data is noiseless, in the presence of even small amounts of noise these algorithms fail.

The problem becomes much harder when the input sequence contains multiple objects with different motions. Not only do we need to recover camera parameters and scene geometry, but we also need to decide which data points should be grouped together. This problem was formulated as a matrix factorization problem by Costeira-Kanade [3]. They suggested to compute an affinity matrix related to the singular value

decomposition of the measurements matrix. Then they decide whether two points have the same motion or not by inspecting if some entries of this affinity matrix are zero or not (Gear [5] and Zelnik-Manor et al. [17] follow a similar approach). In the noiseless case these methods perform well, but once even small amounts of noise exist, these methods no longer work since matrix entries that were supposed to be zero are not zero anymore. Furthermore, these methods require some prior knowledge on the rank of the different motions, or linear independence between them.

In this paper we present a framework for matrix factorization capable of incorporating priors and enforcing strict constraints on the desired factorization while handling missing data and correlated noise in the observations. Previous versions of this work were published in [7, 8].

2 Structure From Motion: Problem Formulation and an Algorithm

A set of P feature points in F images are tracked along an image sequence. Let (u_{fp}, v_{fp}) denote image coordinates of feature point p in frame f . Let $W = (w_{ij})$ where $w_{2i-1,j} = u_{ij}$ and $w_{2i,j} = v_{ij}$ for $1 \leq i \leq F$ and $1 \leq j \leq P$.

In the orthographic camera model, points in the 3D world are projected in parallel onto the image plane. For example, if the image coordinate system is aligned with the coordinate system of the 3D world, then a point $P = [X, Y, Z]^T$ is projected to $p = (u, v) = (X, Y)$ (the depth, Z , has no influence on the image). In this model, a camera can undergo rotation, translation, or a combination of the two. W can be written as [15]:

$$[W]_{2F \times P} = [M]_{2F \times 4} [S]_{4 \times P} + [\eta]_{2F \times P} \quad (1)$$

where $M = \begin{bmatrix} M_1 \\ \vdots \\ M_F \end{bmatrix}_{2F \times 4}$ and $S = \begin{bmatrix} X_1 & \cdots & X_P \\ Y_1 & \cdots & Y_P \\ Z_1 & \cdots & Z_P \\ 1 & \cdots & 1 \end{bmatrix}_{4 \times P}$.

Each M_i is a 2×4 matrix that describes camera parameters in the i 'th frame. It consists of location and orientation $[M_i]_{2 \times 4} = \begin{bmatrix} m_i^T & d_i \\ n_i^T & e_i \end{bmatrix}$ where m_i and n_i are 3×1 vectors that describe the rotation of the camera; d_i and e_i are scalars describing camera translation¹. The matrix S contains the 3D coordinates of the feature points, and η is Gaussian noise.

If the elements of the noise matrix η are uncorrelated and of equal variance then we seek a factorization that minimizes the mean squared error between W and MS . This can be solved trivially using the SVD of W . Missing data can be modeled using equation 1 by assuming some elements of the noise matrix η have infinite variance. Obviously SVD is not the solution once we allow different elements of η to have different variances.

¹ Note that we do not subtract the mean of each row from it, since in case of missing data the centroids of visible points in different rows of the matrix do not coincide.

2.1 Factorization as factor analysis

We seek a factorization of W to M and S that minimizes the weighted squared error $\sum_t [(W_t - M_t S)^T \Psi_t^{-1} (W_t - M_t S)]$, where Ψ_t^{-1} is the inverse covariance matrix of the feature points in frame t .

It is well known that the SVD calculation can be formulated as a limiting case of maximum likelihood (ML) factor analysis [12]. In standard factor analysis we have a set of observations $\{y(t)\}$ that are linear combinations of latent variables $\{x(t)\}$:

$$y(t) = Ax(t) + \eta(t) \quad (2)$$

with $x(t) \sim N(0, \sigma_x^2 I)$ and $\eta(t) \sim N(0, \Psi_t)$. In the case of a diagonal Ψ_t with constant elements $\Psi_t = \sigma^2 I$ then in the limit $\sigma/\sigma_x \rightarrow 0$ the ML estimate for A will give the same answer as the SVD.

Let $A = S^T$. Identifying $y(t)$ with the t 'th row of the matrix W and $x(t)$ with the t 'th row of M , then equation 1 is equivalent (transposed) to equation 2. Therefore, equation 1 can be solved using the EM algorithm for factor analysis [13] which is a standard algorithm for finding the ML estimate for the matrix A . The EM algorithm consists of two steps: (1) the expectation (or E) step in which expectations are calculated over the latent variables $x(t)$ and (2) the maximization (or M) step in which these expectations are used to maximize the likelihood of the matrix A . The updating equations are:

E step:

$$E(x(t)|y(t)) = (\sigma_x^{-2} I + A^T \Psi_t^{-1} A)^{-1} A^T \Psi_t^{-1} y(t) \quad (3)$$

$$V(x(t)|y(t)) = (\sigma_x^{-2} I + A^T \Psi_t^{-1} A)^{-1} \quad (4)$$

$$\langle x(t) \rangle = E(x(t)|y(t)) \quad (5)$$

$$\langle x(t)x(t)^T \rangle = V(x(t)|y(t)) + \langle x(t) \rangle \langle x(t) \rangle^T \quad (6)$$

Although in our setting the matrix A must satisfy certain constraints, the E-step (in which the matrix A is assumed to be given from the M-step) remains the same as in standard factor analysis. So far, we assumed no prior on the motion of the camera, i.e. $\sigma_x \rightarrow \infty$ and thus $\sigma_x^{-2} \rightarrow 0$. In subsection 2.2 we describe how to incorporate priors regarding the motion into the E-step.

M step: In the M step we find the 3D coordinates of a point p denoted by $s_p \in R^3$:

$$s_p = B_p C_p^{-1} \quad (7)$$

where

$$\begin{aligned} B_p &= \sum_t [\Psi_t^{-1}(p, p)(u_{tp} - \langle d_t \rangle) \langle m(t)^T \rangle \\ &\quad + \Psi_t^{-1}(p + P, p + P)(v_{tp} - \langle e_t \rangle) \langle n(t) \rangle^T] \\ C_p &= \sum_t [\Psi_t^{-1}(p, p) \langle m(t)m(t)^T \rangle \\ &\quad + \Psi_t^{-1}(p + P, p + P) \langle n(t)n(t)^T \rangle] \end{aligned} \quad (8)$$

where the expectations required in the M step are the appropriate subvectors and submatrices of $\langle x(t) \rangle$ and $\langle x(t)x(t)^T \rangle$.

If we set $\Psi_t^{-1}(p, p) = 0$ when point p is missing in frame t then we obtain an EM algorithm for factorization with missing data. Note that the form of the updates means that we can put any value we wish in the missing elements of y and they will be ignored by the algorithm.

A more realistic noise model for real images is that Ψ_t is *not diagonal* but rather that the noise in the horizontal and vertical coordinates of the same point are correlated with an arbitrary 2×2 inverse covariance matrix. It can be shown that the posterior inverse covariance matrix is $\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$ (I_x and I_y are the directional derivatives of the image and the sum is taken over a window of fixed size around each pixel). This problem is usually called *factorization with uncertainty* [9, 11]. To consider dependencies between the u and v coordinates of a point, the matrix W can be reshaped (to size $F \times 8$) to have both coordinates in the same row (with a corresponding change in M and S). A non diagonal Ψ_t would express the correlation of the noise in the horizontal and vertical coordinates of the same point. With this representation, it is easy to derive the M step in this case as well. It is similar to equation 7 except that cross terms involving $\Psi_t^{-1}(p, p + P)$ are also involved:

$$s_p = (B_p + B'_p)(C_p + C'_p)^{-1} \quad (9)$$

where

$$\begin{aligned} B'_p &= \sum_t [\Psi_t^{-1}(p, p + P)(v_{tp} - \langle e_t \rangle) \langle m(t)^T \rangle \\ &\quad + \Psi_t^{-1}(p + P, p)(u_{tp} - \langle d_t \rangle) \langle n(t)^T \rangle] \\ C'_p &= \sum_t [\Psi_t^{-1}(p, p + P) \langle n(t)m(t)^T \rangle \\ &\quad + \Psi_t^{-1}(p + P, p) \langle m(t)n(t)^T \rangle] \end{aligned} \quad (10)$$

Regardless of uncertainty and missing data, the complexity of the EM algorithm grows linearly with the number of feature points and the number of frames.

2.2 Adding Priors on the desired factorization

The EM framework allows us to place priors on both structure and motion and to deal with directional uncertainty and missing data. We first show how to place a prior on the motion in the form of temporal coherence. Next we show how to place a prior on the 3D structure of the scene.

Temporal coherence: The factor analysis algorithm assumes that the latent variables $x(t)$ are independent (figure 1(a)). In SFM this assumption means that the camera locations in different frames are independent and hence permuting the order of the frames

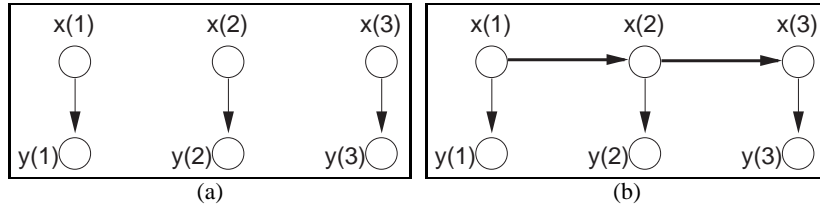


Fig. 1. a. The graphical model assumed by most factorization algorithms for SFM. The camera location $x(t)$ is assumed to be independent of the camera location at any other time step. **b.** The graphical model assumed by our approach. We model temporal coherence by assuming a Markovian structure on the camera location.

makes no difference for the factorization. In almost any video sequence this assumption is wrong. Typically camera location varies smoothly as a function of time (figure 1(b)). Specifically, in this work we use a second order approximation to the motion of the camera (details can be found in [7, 8]). Note that we do not assume that the 2D trajectory of each point is smooth. Rather we assume the 3D trajectory of the camera is smooth.

It is straightforward to derive the EM iterations for a ML estimate of S using the model in figure 1(b). The M step is unchanged from the classical factor analysis. The only change in the E step is that $E(x(t)|y)$ and $V(x(t)|y)$ need to be calculated using a Kalman smoother. We use a standard RTS smoother [6]. Note that the computation of the E step is still linear in the number of frames and datapoints.

Prior on Structure: Up to this point, we have assumed nothing regarding the 3D coordinates of the feature points we are trying to reconstruct. The true 3D coordinates are considered (a priori) as likely as any other coordinates, even ones that suggest the object is located at an infinite position, or behind the camera, for example. Usually when sequences are acquired for structure reconstruction, the object is located just in front of the camera in the center of the scene, and not at infinity². Therefore, we should prefer reconstructions that place the feature points around certain coordinates in the world, denoted by S_0 (typically X and Y are scattered around zero and Z is finite). We model this preference with the following prior: $\Pr(S) \propto e^{-\lambda \|S - S_0\|_F^2}$, where λ is a parameter that determines the weight of this prior.

Derivation of the modified M-step with the addition of the prior on structure yields (the following modification of equation 9):

$$s_p = (B_p + B'_p)(C_p + C'_p + \lambda(I - S_0))^{-1} \quad (11)$$

Experimental results show an improvement in reconstruction results in noisy scenes after the addition of this naive prior.

² Although for objects to comply with affine model they have to be located relatively far from the camera, they are not placed at infinity.

3 Constrained Factorization for Subspace Separation

In dynamic scenes with multiple moving objects, each of the K independent motions has its own motion parameters, M_i^j (a 2×4 matrix describing the j th camera parameters at time i). Denote by S_j the $4 \times P_j$ matrix of the P_j points moving according to the j th motion component. The matrix formed by taking the locations of points sharing the same motion along the sequence is of rank 4. In other words, the vectors of point locations of points with the same motion form a 4D linear subspace defined by the common motion (in fact this is a 3D affine subspace). This is a problem of subspace separation.

Let \tilde{W} be a matrix of observations whose columns are ordered to group together points with the same motion. Then ([3]):

$$[\tilde{W}]_{2F \times P} = M\tilde{S} = \begin{bmatrix} M_1^1 & \cdots & M_1^K \\ \vdots & & \vdots \\ M_F^1 & \cdots & M_F^K \end{bmatrix}_{2F \times 4K} \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ 0 & S_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & S_K \end{bmatrix}_{4K \times P} \quad (12)$$

In real sequences, however, measurements are not grouped according to their motion. Therefore, the observation matrix, W , is an arbitrary column permutation of the ordered matrix \tilde{W} :

$$W = \tilde{W}II = MS \quad (13)$$

where $S_{4K \times P}$ describes scene structure (with unordered columns) and $II_{P \times P}$ is a column permutation matrix. Hence, the structure matrix S is in general not block diagonal, but rather a column permutation of a block diagonal matrix:

$$S = \tilde{S}II \quad (14)$$

Therefore, in each column of the structure matrix corresponding to a point belonging to the k th motion, only entries $4(k-1) + 1, \dots, 4k$ can be non-zeros (entry $4k$ always equals one).

Finding a factorization of W to M and S that satisfies this constraint would solve the subspace separation problem: from the indices of the non-zero entries in S we can assign each point to the appropriate motion component.

The constrained factorization problem can be written as a constrained factor analysis problem as follows: By substituting $A = S^T$ and identifying $x(t)$ with the t th row of M , the constrained factorization problem is equivalent to the factor analysis problem of equation 2 where A is subject to the constraints on S^T . We adapt the EM algorithm for single motion presented in the previous section to solve constrained factor analysis problem.

Since the matrix A is assumed to be known in the E step, no change is required in the E step of the algorithm from the previous section. The M step, on the other hand, should be modified to find A that satisfies the constraints.

We modify the M step to find S that is a permuted block diagonal matrix. The columns of S (which are the rows of A) can be found independently on each other

(each point is independent on the other points given the motion). We show how to find each of the columns of S that will contain non zeros only in the 4 entries corresponding to its most likely motion. Denote by π_p the motion that maximizes the likelihood for point p and let $\pi = (\pi_1, \dots, \pi_P)$. Let s_p denote the 3D coordinates of point p , and let \hat{S} denote $[s_1, \dots, s_P]$ the 3D coordinates of all points (S contains both segmentation and geometry information, \hat{S} contains only geometry information).

We look for S that maximizes the expected complete log likelihood (where the expectation is taken over M , the motion parameters of all motion components at all times). Maximizing the expected complete log likelihood is equivalent to minimizing of the expectation of an energy term. In terms of energy minimization, the expectation of the energy due to equation 13 is:

$$\begin{aligned} E(S) = E(\hat{S}, \pi) &= \left\langle E(\hat{S}, \pi, M) \right\rangle_M = \\ &= \sum_p \langle E(s_p, \pi_p, M) \rangle_M = \\ &= \sum_p \sum_t \langle \langle (W_{t,p} - M_{t,\pi_p} s_p)^T \Psi_{t,p}^{-1} (W_{t,p} - M_{t,\pi_p} s_p) \rangle \rangle_M \end{aligned} \quad (15)$$

The energy is the weighted sum of square error of the matrix equation 13. In other words, it is the sum of the error over all the points at all times, weighted by the inverse covariance matrix $\Psi_{t,p}^{-1}$ (the sum over the points is implicit in the vectorial notation of the energy for a single motion at the beginning of section 2.1).

As can be seen from equation 15, $E(S)$ can be represented as a sum of terms $E_p(s_p, \pi_p) = \langle E(s_p, \pi_p, M) \rangle_M$ involving a single point:

$$E(S) = \left\langle E(\hat{S}, \pi, M) \right\rangle_M = \sum_p E_p(s_p, \pi_p) \quad (16)$$

Therefore the minimization of $E(S)$ can be performed by minimizing $E_p(s_p, \pi_p)$ for each point p independent on the others.

Since s_p is unknown, we define

$$E_p(\pi_p) = \min_{s_p} E_p(s_p, \pi_p) \quad (17)$$

And we get

$$\min_{s_p, \pi_p} E_p(s_p, \pi_p) = \min_{\pi_p} \left[\min_{s_p} E_p(s_p, \pi_p) \right] = \min_{\pi_p} E_p(\pi_p) \quad (18)$$

Let $s_p^k = \arg \min_{s_p} E_p(s_p, k)$ for a given k . The value of s_p^k can be computed using one of the equations 7, 9,11, replacing $d_t, m(t), e_t$ and $n(t)$ with $d_t^k, m_k(t), e_t^k$ and $n_k(t)$ respectively. Once all the s_p^k are known, $E_p(k)$ are computed for all k by substituting s_p^k in equation 15. Then we choose $\pi_p = \arg \min_k E_p(k)$. The new value of the p th column of S is all zeros except the four entries $4(\pi_p - 1) + 1, \dots, 4\pi_p$. Entries $4(\pi_p - 1) + 1, \dots, 4(\pi_p - 1) + 3$ are set to be s_p^k and entry $4\pi_p$ is set to 1.

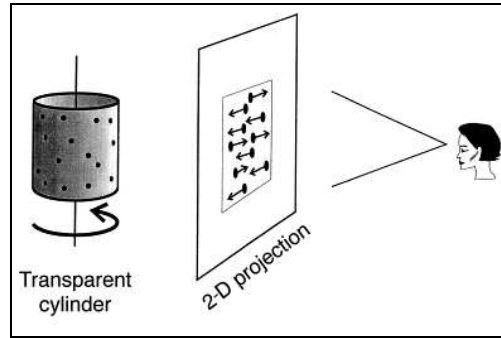


Fig. 2. Synthetic input for evaluation of structure from motion algorithms. A transparent cylinder is rotating around its elongated axis. Points randomly drawn from its surface are projected on the camera plane at each frame. Replotted from [1].

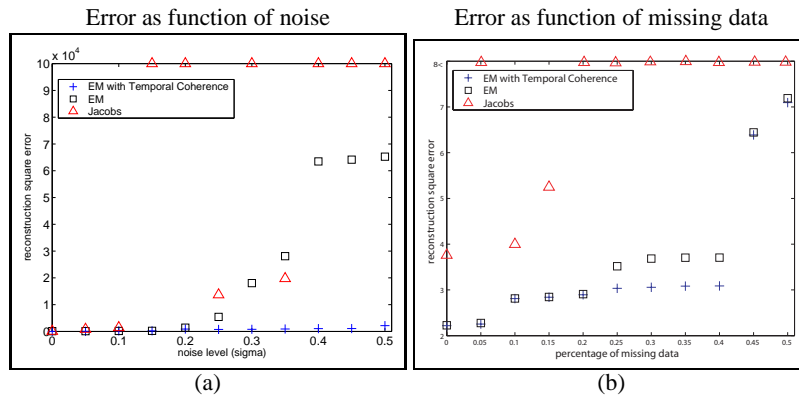


Fig. 3. The graphs depict influence of noise and percentage of missing data on reconstruction results of factor analysis and [10]. The input sequence for these experiments is depicted in figure 2.

By modifying the EM algorithm to deal with constrained factorization we now have an algorithm that is guaranteed to find a factorization where the structure matrix has at most 4 nonzero elements per column, even in the presence of noise (in contrast to [3, 5, 17]). Note that no prior knowledge of the rank of the different motions is needed, neither is any assumption on the linear independence of the different motions.

4 Experiments

In this section we describe the experimental performance of EM for SFM and for motion segmentation. In each case we describe the performance of EM with and without temporal coherence.

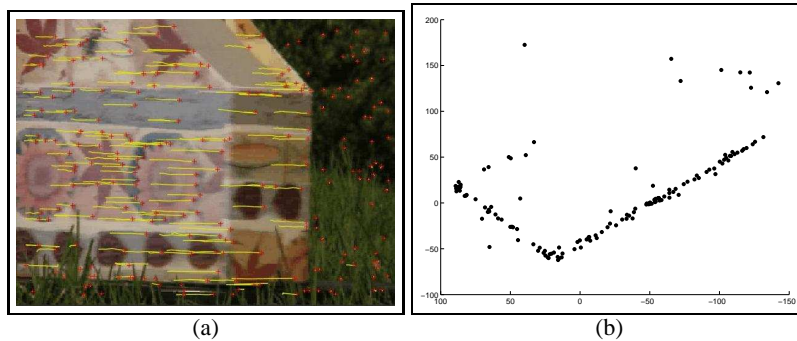


Fig. 4. Results of scene reconstruction from a real sequence: A binder is placed on a rotating surface filmed with a static camera. Our algorithm succeeded in (approximately) obtaining the correct structure while all other algorithms failed. **a.** The first frame of the sequence. **b.** The reconstructed object shown in top view. The 3 lines visible are the outlines of the object. Each of these lines is the vertical projection of each of the 3 visible sides of the box. The longer line corresponds to the side of the box closer to the camera and the shorter lines correspond to the 2 other sides visible along the sequence.

4.1 EM for SFM

We evaluate EM for structure from motion compared to ground truth and to previous algorithms for structure from motion with missing data [15, 10, 14, 2]. For [15, 10, 14] we used the Matlab implementation made public by D. Jacobs³.

The first input sequence is a synthetic sequence of a transparent rotating cylinder as depicted in figure 2. This sequence (that was first presented in [16]) consists of 100 points uniformly drawn from the cylinder surface. The points are tracked along 20 frames. We checked the performance of the different algorithms in the following cases: (1) full noise free observation matrix, (2) noisy full observation matrix (to create noisy input, the observed image locations were added a Gaussian noise with $\sigma = 0, \dots, 0.5$), (3) noiseless observations with missing data and (4) noisy observations with missing data.

All algorithms performed well and gave similar results for the full matrix noiseless sequence.

In the fully observed noisy case, factor analysis without temporal coherence gave comparable performance to the algorithm of Tomasi-Kanade, which minimizes $\|MS - W\|_F^2$. When temporal coherence was added, the reconstruction results were improved. The results of Shum's algorithm were similar to Tomasi-Kanade. The algorithms of Jacobs and Brand turned out to be noise sensitive.

In the experiments with missing data, Tomasi-Kanade's algorithm and Shum's algorithm could not handle this pattern of missing data and failed to give any structure. The algorithms of Jacobs and Brand turned out to be noise sensitive.

Figure 3 shows a comparison between both versions of EM and the algorithm of Jacobs. The performance of the algorithms was tested as a function of noise level and

³ The code is available at <http://www.cs.umd.edu/~djacobs>

percentage of missing data. Figure 4 shows result on a real sequence for which EM with temporal coherence succeeded to recover the correct structure, while all other algorithms have failed.

4.2 EM for Motion Segmentation

Figure 5 shows quantitative comparisons of EM and Costeira and Kanade for three different synthetic sequences as a function of noise level. It is apparent that all algorithms give perfect segmentation when there is no noise at all. As the amount of noise increases, the performance of [3] deteriorates rapidly, while EM-based segmentation continues to succeed for low amounts of noise and shows moderate increase in the number of errors for larger amounts of noise. It is also clear that EM with temporal coherence performs significantly better than EM without temporal coherence for noisy inputs. The algorithms of [5, 17] perform similar to [3] in non-degenerate cases when the actual rank of observation matrix is provided.

Figure 6 shows the performance of EM with temporal coherence as a function of the percentage of missing data. While all other factorization algorithms cannot work with missing data, EM continues to perform well even when 50% of the data is missing. For comparison, we also show the algorithm of [3] when the observation matrix is first filled in using Jacobs' algorithm [10] and the correct rank is given to all algorithms.

Finally, we tested the different algorithms on a real sequence of two cans rotating horizontally around parallel different axes in different angular velocities. 149 feature points were tracked along 20 frames, from which 93 are from one can, and 56 are from the other. Some of the feature points were occluded in part of the sequence, due to the rotation. Notice that despite of its simple appearance, this is a rather challenging scene because a large percentage of the points are missing and because of the motion degeneracy: the two cans have "similar" motion, that is rotation around parallel axes, which leads to a rank deficient motion matrix.

Using EM for motion segmentation, 8 points were misclassified. For comparison, Costeira-Kanade (using the maximal full submatrix of the measurements matrix) resulted in 30 misclassified points and a failure in 3D structure reconstruction. Figure 7(a) shows the first frame of the sequence and the tracks superimposed and figures 7(b), 7(c) show the curved surface of the two cylinders recovered correctly.

5 Discussion

In this paper we have presented an EM algorithm for matrix factorization based on representing the factorization problem as a problem of factor analysis.

Working with this representation allowed us to (1) handle correlated measurements noise and missing data, (2) place informative priors on both structure and motion enabling 3D reconstruction in scenes where previous methods have failed and (3) impose constraints on the resulting factors, thereby extending the applicability of factorization methods to problems such as subspace separation.

It would be interesting to study applications of the enhanced factorization capabilities presented in this paper in other vision problems and in problems taken from other areas, for example, semantic analysis of texts ([4]).

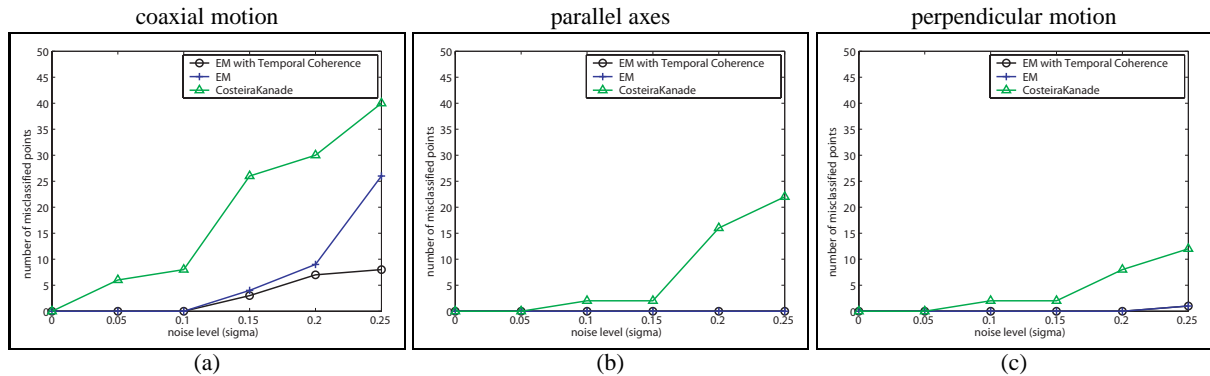


Fig. 5. Comparison of different factorization algorithms for motion segmentation on synthetic inputs. The graphs display total number of misclassified points as a function of the noise standard deviation for $\sigma = 0, \dots, 0.25$. In some of the experiments, the graphs of the two factor analysis versions overlap. **a.** sequence of concentric cylinders rotating in different speeds. Due to the input degeneracy only EM and [3] are compared. **b.** a cylinder and a cube rotating in the same speed around different parallel axes. **c.** A cube and a cylinder rotating around perpendicular axes.

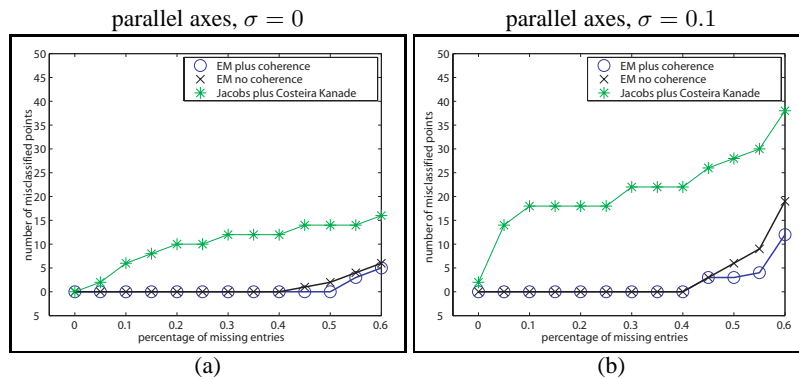


Fig. 6. Performance of EM for motion segmentation with and without temporal coherence. Graphs show number of misclassifications as a function of the percentage of missing data. **a.** Cube and cylinder rotating around different parallel axes without noise. **b.** Cube and cylinder rotating around different parallel axes with noise with standard deviation $\sigma = 0.1$.

References

1. R.A. Andersen and D.C Bradley. Perception of three-dimensional structure from motion. In *Trends in Cognitive Sciences*, 2, pages 222–228, 1998.
2. M.E. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–720, May 2002.
3. J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1071–1076, 1995.

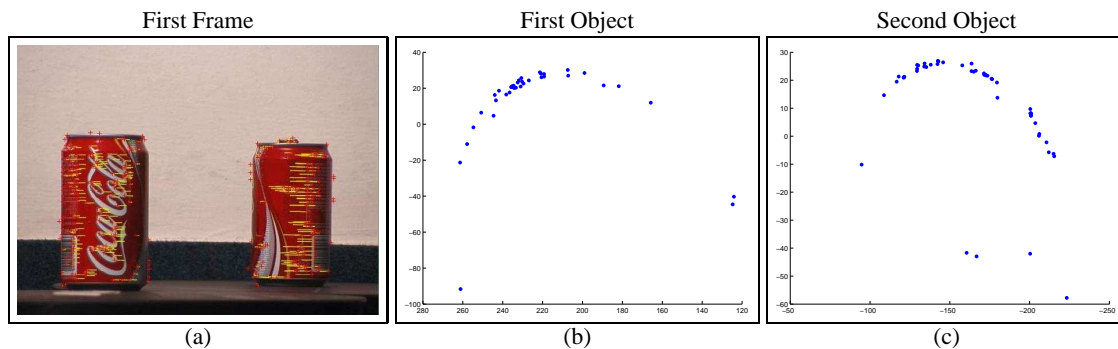


Fig. 7. A real sequence of two cans rotating around different parallel axes. EM with temporal coherence succeeds in finding correct segmentation and 3D structure reconstruction while other existing algorithms fail. See text for further details. **a.** First image from the input sequence with tracks found by tracking software superimposed. **b.** First segment, top view. **c.** Second segment, top view.

4. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
5. C.W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision (IJCV)*, pages 133–150, 1998.
6. A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.
7. A. Gruber and Y. Weiss. Factorization with uncertainty and missing data: Exploiting temporal coherence. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2003.
8. A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2004.
9. M. Irani and P. Anandan. Factorization with uncertainty. In *Proceedings of the European Conference on Computer Vision (ECCV) (1)*, pages 539–553, 2000.
10. D. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 206–212, 1997.
11. D. D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertain models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 696–702, January 1999.
12. S. Roweis. EM algorithms for PCA and SPCA. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 431–437, 1997.
13. D. Rubin and D. Thayer. EM algorithms for ML factor analysis. *Psychometrika* 47(1), pages 69–76, 1982.
14. H. Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 854–867, September 1995.
15. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision (IJCV)*, 9(2):137–154, November 1992.
16. S. Ullman. *The interpretation of visual motion*. MIT Press, 1979.
17. L. Zelnik-Manor, M. Machline, and M. Irani. Multi-body segmentation: Revisiting motion consistency. In *Workshop on Vision and Modeling of Dynamic Scenes, with ECCV*, 2002.