Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-3-2004

An Iterative Approach to Recovering the Missing Data in a Large Low-rank: Application to SFM

P. Chen and D. Suter

MONASH
UNIVERSITY

Recovering the Missing Components in a Large Noisy Low-Rank Matrix: Application to SFM

Pei Chen and David Suter
Dept ECSE, Monash University, Australia, 3800
{pei.chen, david.suter}@eng.monash.edu.au

Abstract

In the field of computer vision, it is common to require operations on matrices with "missing data", for example because of occlusion or tracking failures. In this paper, we consider a special case, where the large matrix should be of low rank if it is noise free. This constraint often exists, such as in the factorization method for the problem of structure from motion (SFM). In this paper, we propose a new iterative solution method to the missing-data problem. It has the following advantage: (i) Fast convergence. (ii) The recoverability of the unknown entries can be easily determined. (iii) The initial result, after the initialization step, is exactly correct and no iteration step is required if the data available is noise free and the incomplete matrix is recoverable. We compare the performance of the proposed method with Jacobs' method. The iterative algorithm performs much better than Jacobs' method when applied to both synthetic data and real data. Moreover, even after merely the initialization step, the proposed method usually exhibits a better performance than Jacobs' method.

1 Introduction

Several problems in computer vision can be reduced to fitting a large matrix to its closest low-rank approximation, such as the factorization method under affine models [6,7,9,13], and optical flow estimation in multi-frame video [2,3]. In the non-degenerate cases, the measurement matrix, consisting of the feature points, should be exactly of rank 4 [6,7,9,13]. (Although the registered measurement matrix should be of rank 3 [6,7,9,13].) However, noise is inevitably introduced in the data: such as that due to the distortion in the imaging process, or quantization error, or even due to wrong matching of the feature points. In the presence of noise, the matrix quickly becomes full-rank. Thus, the matrix has to be fitted to its low-rank approximation. The singular value decomposition (SVD) gives the best solution to this problem [1]. However, SVD works only when all the entries in the large matrix are available. This requirement is so strong that it has been regarded [5,10] as the major drawback of the factorization method, although Tomasi and Kanade have addressed this in their method [13]. In that somewhat heuristic approach, a full submatrix is first decomposed by the factorization method, then the initial solution grows by one row or by one column at a time. The final estimate is then refined by employing a steepest descent minimization method.

Very little other work has been done to address the missing-data problem, although it is very common, until Shum et al. [11] and Jacobs [4,5]. Jacobs [4,5] treated each column, with some entries unknown, as an affine subspace, and solved the problem by obtaining the intersection of all the triple affine subspaces (in principle, in practice a selection is used). The greatest advantage of Jacobs' method lies in the fact that it does not need to start from a complete submatrix. Jacobs [4,5] also suggested that Shum's technique [11] could be applied to the problem of recovering the missing data in SFM, though, strictly speaking, that paper [11] did not directly address this problem. Recently, by combining Jacobs' method [4,5] with the projective factorization method of Sturm & Triggs [12], Martinec et al. [8] solved the missing-data problem under the perspective model. Other geometric constraints [6,10], than the subspace constraint, were also employed to cope with the missing-data problem.

In this paper, we propose a new iterative approach to the problem of low rank approximation in the presence of missing data. The method starts by the minimization of the distance of a vector, with some unknown entries, to a known subspace. Then, from this, we iteratively refine the unknown entries. In terms of the starting point, our approach is similar to Tomasi and Kanade's approach [13] to the problem because our approach also needs to start from a complete submatrix. This is potentially a major drawback, as criticized by Jacobs [4,5]. However, it always converges to the global minimum when the noise level is not too strong or when the percentage of the missing data is not very high. So, the proposed method does not strongly depend on the initial submatrix, as will be proved by many synthetic-data experiments and some results using real data. For example, for a rank-4 matrix, the method yields a good result by starting from an 8-by-8 submatrix, which means that only 8 points in 4 frames are required as a starting point. In the severe cases, where the proposed method fails due to strong noise and too many missing entries, Jacobs' method often produces an unsatisfactory estimate. Yet, for the proposed method, it is very easy to determine whether the missing entries can be recovered. The cases that can not be recovered were regarded as *unstable* in Jacobs' method; however, no approach was presented in [4,5] to determine whether the incomplete matrix is *stable* or *unstable*.

Another characteristic of the proposed method is that the known data is untouched in the first stage recovery process. So, the recovery step in the proposed method is completely separated from other steps, like the factorization in SFM problem. This contrasts with Jacobs' method, where the matrix is exactly of low rank immediately after the recovery.

It is worth noting that the result after the initial recovery step in our method, is exactly correct (and no iteration step is required) if the data available is noise free and the incomplete matrix is recoverable.

In section 2, we first state the problem, using an objective function that is subtly different from the one in Shum's method. Then, we propose a new iterative method of recovering the missing data in a large low-rank matrix. In section 3, we present the justification of the algorithm. The discussion includes three parts: the principle behind recovering the unknown entries so that the distance of the vector to a known subspace is minimized; assessing the possibility of recovering the unknown entries; and the convergence of the iterative algorithm. In section 4, we compare the algorithm with Jacobs' method on synthetic data and real data.

2 The definition of the problem and an iterative algorithm solving the problem

2.1 Notation

In the following, a matrix will be denoted by a bold capital letter, like $\mathbf{M}$, and a bold lowercase letter represents a vector, e.g. $\mathbf{x}$. A scalar entry in a vector or in a matrix will respectively be denoted by, for example, x_1 or $M_{1,2}$. $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. A matrix $\mathbf{U}$, $\mathbf{U} \in R^{m,n}$, is said to be orthonormal, iff $\mathbf{U}^T\mathbf{U} = \mathbf{I}_n$. The set of $m \times n$ orthonormal matrices is denoted by $O^{m,n}$. An orthonormal matrix will always be denoted by $\mathbf{U}$ or $\mathbf{V}$. The Frobenius norm of a matrix $\mathbf{M}$ (or a vector) will be denoted as $\left\|\mathbf{M}\right\|_F$, where $\left\|\mathbf{M}\right\|_F = \sqrt{\sum_{i,j} M_{i,j}^2}$. $span(\mathbf{M})$ denotes the subspace spanned by the columns of $\mathbf{M}$. The distance of a vector $\mathbf{m}$, $\mathbf{m} \in R^m$, to the subspace $span(\mathbf{M})$, $\mathbf{M} \in R^{m,n}$, is denoted as $d(\mathbf{m}, span(\mathbf{M}))$ and it is sometimes described as the distance of a vector $\mathbf{m}$ to a matrix $\mathbf{M}$, denoted as $d(\mathbf{m},\mathbf{M})$. If the matrix, $\mathbf{U}$, is orthonormal, $d(\mathbf{m},\mathbf{U}) = \left\|\mathbf{m} - \mathbf{U}\mathbf{U}^T\mathbf{m}\right\|_F$. Similarly, we can define the distance of a matrix $\mathbf{N}$ to another matrix $\mathbf{M}$. The distance of a matrix $\mathbf{M} \in R^{m,n}$ to an orthonormal matrix $\mathbf{U} \in O^{m,r}$ is $d(\mathbf{M},\mathbf{U}) = \left\|\mathbf{M} - \mathbf{U}\mathbf{U}^T\mathbf{M}\right\|_F$. The hat symbol, "^", denotes an estimate of the quantity beneath the symbol. $\mathbf{M}^r$ denotes the closest rank-*r* approximation of $\mathbf{M}$, which can be obtained by SVD [1]: $\mathbf{M}^r = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in O^{m\times r}$, $\mathbf{\Sigma} \in R^{r\times r}$ and $\mathbf{V} \in O^{n\times r}$.

2.2 The problem

A large matrix $\mathbf{M} \in R^{m,n}$, which should have a low rank r, is corrupted with some noise, usually assumed to be i.i.d. Gaussian, and has some unknown entries. The problem is to recover these missing entries and to make the recovered matrix as close as possible to one rank-*r* matrix. Analytically, the objective is to minimize the distance between the recovered matrix, $\hat{\mathbf{M}}$, and its closest rank-*r* approximation, $\hat{\mathbf{M}}^r$:

$$\min \quad d(\hat{\mathbf{M}}, \hat{\mathbf{M}}^r) \tag{1}$$

subject to $\hat{M}_{i,j} = M_{i,j}$ if $M_{i,j}$ is observed.

Note: The minimization objective in the problem above is a little different from that in Shum's approach, where the objective is to minimize the sum of the square of the difference between known elements in the incomplete matrix and the corresponding elements in the new matrix, which is exactly of low-rank.

2.3 Non-linearity of the problem

Here, by examining the singular values of the matrix, we provide a brief discussion concerning the intrinsic non-linearity of the problem above. Suppose $\mathbf{M} \in R^{m,n}$. Its closest rank-*r* matrix, measured by the Frobenius norm, is $\mathbf{M}^r = \mathbf{U}^r\mathbf{\Sigma}^r(\mathbf{V}^r)^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$, and the distance is $\left\|\mathbf{M} - \mathbf{M}^r\right\|_F = \sum_{i=r+1}^{p} \sigma_i^2$ [1], where $p = \min(m,n)$ and $\{\sigma_i^2\}$ are the non-descending eigenvalues of $\mathbf{M}^T\mathbf{M}$.

Suppose $\mathbf{M}$ has some unknown entries $\{M_{i,j} \mid (i,j) \in \Xi\}$, where $\Xi = \{(i,j) \mid M_{i,j} \text{ is unknown}, 1 \le i \le m, 1 \le j \le n\}$. $\mathbf{E}_{i,j} \in R^{m,n}$, all of whose entries are zeros, except $E_{i,j} = 1$. Let the recovered matrix be $\hat{\mathbf{M}}$, $\hat{\mathbf{M}} = \overline{\mathbf{M}} + \sum_{(i,j)\in\Xi} k_{i,j}\mathbf{E}_{i,j}$, where $\overline{M}_{i,j} = \begin{cases} M_{i,j} & (i,j) \notin \Xi \\ 0 & (i,j) \in \Xi \end{cases}$. The characteristic polynomial of $\hat{\mathbf{M}}^T\hat{\mathbf{M}}$, $p(\lambda)$, is a high-order polynomial of λ and $k_{i,j}$. The equation, $p(\lambda) = 0$, has n non-negative roots for any $\{k_{i,j}\}$, because $\hat{\mathbf{M}}^T\hat{\mathbf{M}}$ is positive semi-definite. The problem reduces to finding $\{\hat{k}_{i,j}\}$, which minimizes the sum of the least $n-r$ roots of the equation, $p(\lambda) = 0$. This is a nonlinear problem.

Specifically, we consider a simple case, where the matrix is $\mathbf{M} \in R^{10,10}$ and has an unknown entry $M_{1,1}$. Suppose $\mathbf{M}$ should be of rank-4, if it were noise free and had no unknown entries. With an unknown entry, its characteristic polynomial, $p(\lambda, t)$, where t denotes the unknown entry, is of the form:

$$p(\lambda,t) = \lambda^{10} + f_2(\lambda)t^2 + f_1(\lambda)t + f_0(\lambda) = \lambda^{10} + \sum_{i=0}^{9} \lambda^i g_i(t)$$

where $f_i(\lambda) = \sum_{j=0}^{j=9} f_{i,j}\lambda^j$ and $g_i(t) = \sum_{j=0}^{j=2} g_{i,j}t^j$, and $f_{i,j}$ and $g_{i,j}$ are determined by $\mathbf{M}$. This equation is nonlinear and the problem of minimizing the sum of the least 6 roots becomes very complicated. If there are many unknown entries in the matrix, the problem becomes intractable from this point of view.

2.4 An iterative algorithm for the problem

Because the problem, in section 2.2, is intrinsically nonlinear, no analytical solution exists. In this subsection, we present an iterative algorithm to the problem above. We defer the justification of this algorithm until section 3.

Algorithm 1

(i) Searching for a complete submatrix: Suppose, without loss of generality, the matrix $\mathbf{M}$, after some row exchanges and column exchanges, has a block representation as: $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$, where all entries in $\mathbf{A}$ are known, and some entries in $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are unknown.

(ii) Initialization: *(a)*. First consider the submatrix $[\mathbf{A} \quad \mathbf{B}]$. Recover $\hat{\mathbf{B}}$ from $\mathbf{A}$ by theorem 1 or theorem 2, and obtain $\begin{bmatrix} \mathbf{A} & \hat{\mathbf{B}}_1 & \mathbf{B}_2 \\ \mathbf{C} & \mathbf{D}_1 & \mathbf{D}_2 \end{bmatrix}$, where the unknown entries in $\hat{\mathbf{B}}_1$ have been recovered and the unknown entries in $\mathbf{B}_2$ can not been recovered. Note: this induces a split of submatrix $\mathbf{D}$. *(b)*. Similarly, recover $[\mathbf{C} \quad \mathbf{D}_1]$ from $[\mathbf{A} \quad \hat{\mathbf{B}}_1]$, and obtain $\begin{bmatrix} \mathbf{A} & \hat{\mathbf{B}}_1 & \mathbf{B}_2 \\ \hat{\mathbf{C}}_1 & \hat{\mathbf{D}}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix}$.

After sub-step (a), check whether all the unknown entries have been recovered. If so, terminate the initialization step and go to the iteration step; if not, go to sub-step (b). After sub-step (b), check for completion again. If all the entries have been recovered, go to the iteration step. If not, check the following condition: Is the number of the non-recovered entries before sub-step (a) same as that number after sub-step (b)? If so, the unknown entries in $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ cannot be recovered. If the number of non-recovered entries decreases, continue the initialization step (a) by regarding the recovered entries as known.

After this initialization procedure, we obtain a recovered matrix $\hat{\mathbf{M}}_1$, and set $d_0 = \infty$.

(iii) Iteration: From $\hat{\mathbf{M}}_i$, obtain its closest rank-*r* approximation by SVD: $\hat{\mathbf{M}}_i^r = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T$. Compute the error $d_i = \left\|\hat{\mathbf{M}}_i^r - \hat{\mathbf{M}}_i\right\|_F$. If

$$d_{i-1} - d_i < \varepsilon \tag{2}$$

terminate the iteration; else, from $\mathbf{U}_i$, recover the missing entries in $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ by theorem 1 or theorem 2, and obtain $\hat{\mathbf{B}}_{i+1}$, $\hat{\mathbf{C}}_{i+1}$ and $\hat{\mathbf{D}}_{i+1}$. Set $\hat{\mathbf{M}}_{i+1} = \begin{bmatrix} \mathbf{A} & \hat{\mathbf{B}}_{i+1} \\ \hat{\mathbf{C}}_{i+1} & \hat{\mathbf{D}}_{i+1} \end{bmatrix}$.

3 Justification of the algorithm

In this section, we motivate and prove the convergence of the algorithm above. First, we propose how to recover the missing components of a vector so that it has the shortest distance to a known subspace. Then, a feasible criterion about the recoverability of the missing entries is presented. Finally, the convergence of the algorithm is proved.

3.1 Minimization of the distance of a vector with missing entries to a known subspace

Here, we first consider a simple case, where the subspace is assumed known. Let $\mathbf{U} \in O^{m,r}$. A vector $\mathbf{x} \in R^m$ has one missing component, and without loss of generality, we suppose x_1 is unknown. The following theorem provides an approach to recover the unknown component $\hat{x}_1$, which minimizes the distance $d(\mathbf{x}, \mathbf{U})$.

Theorem 1: If $\mathbf{e_1} \in span(\mathbf{U})$, $d(\mathbf{x},\mathbf{U}) \equiv C = d(\bar{\mathbf{x}},\mathbf{U})$, where $\bar{\mathbf{x}} = [0, x_2, x_3, \cdots, x_m]^T$, with all entries, except the first entry, same as those in $\mathbf{x}$. If $\mathbf{e_1} \notin span(\mathbf{U})$, $\hat{x}_1 = r_1/(P_{1,1}-1)$ minimizes $d(\mathbf{x},\mathbf{U})$, where $\mathbf{P} = \mathbf{U}\mathbf{U}^T$ and $\mathbf{r} = \bar{\mathbf{x}} - \mathbf{P}\bar{\mathbf{x}}$.

Proof (i) If $\mathbf{e_1} \in span(\mathbf{U})$, construct $\overline{\mathbf{U}} = [\mathbf{e_1}, \mathbf{U}'] \in O^{m,r}$ by Schmidt orthogonalization of the columns of $\mathbf{U}$. Note $span(\overline{\mathbf{U}}) = span(\mathbf{U})$.

$d(\mathbf{x},\mathbf{U}) = d(\mathbf{x},\overline{\mathbf{U}}) = d(\mathbf{x} - <\mathbf{x},\mathbf{e}_1>\mathbf{e}_1, \mathbf{U}') = d(\bar{\mathbf{x}},\mathbf{U}') \equiv C$.

(ii) Suppose $\mathbf{e_1} \notin span(\mathbf{U})$. Construct an enlarged matrix $\hat{\mathbf{S}} = [\mathbf{U},\mathbf{e_1}]$. From the proof above, $d(\mathbf{x},\hat{\mathbf{S}}) \equiv C$**.** So, for the second part, we only need to prove that $d(\hat{\mathbf{x}},\mathbf{U}) = d(\hat{\mathbf{x}},\hat{\mathbf{U}}) \equiv d(\bar{\mathbf{x}},\hat{\mathbf{U}})$ if $\hat{x}_1 = r_1/(P_{1,1}-1)$, and $d(\mathbf{x},\mathbf{U}) \ge d(\mathbf{x},\hat{\mathbf{U}})$. The latter is trivial (because $\mathbf{U} \subset \hat{\mathbf{U}}$).

Apply Schmidt orthogonalization on the enlarged matrix $\hat{\mathbf{S}}$ and obtain its orthonormal representation $\tilde{\mathbf{U}}$, $\tilde{\mathbf{U}} = [\mathbf{U},\mathbf{u}] \in O^{m,r+1}$, where $\mathbf{u} = (\mathbf{e}_1 - \mathbf{P}\mathbf{e}_1)/\|\mathbf{e}_1 - \mathbf{P}\mathbf{e}_1\|_F = (\mathbf{e}_1 - \mathbf{P}\mathbf{e}_1)/l$, i.e. $\mathbf{u}$ is the unit vector in the direction of the orthogonal component of $\mathbf{e}_1$ to the subspace $\mathbf{U}$.

Note: $l^2 = \|\mathbf{e}_1 - \mathbf{P}\mathbf{e}_1\|_F^2 = \mathbf{e}_1^T(\mathbf{I}_m - \mathbf{P})^T(\mathbf{I}_m - \mathbf{P})\mathbf{e}_1 = 1 - P_{11}$

$$\begin{aligned}\hat{\mathbf{x}} - \mathbf{P}\hat{\mathbf{x}} &= (\mathbf{I}_m - \mathbf{P})(\bar{\mathbf{x}} + [\hat{x}_1,0,0,\cdots,0]^T) \\ &= (\mathbf{I}_m - \mathbf{P})\bar{\mathbf{x}} + (\mathbf{I}_m - \mathbf{P})[\hat{x}_1,0,0,\cdots,0]^T\end{aligned} \tag{3}$$

Similarly, define the projection matrix $\tilde{\mathbf{P}}$ upon $\tilde{\mathbf{U}}$, i.e., $\tilde{\mathbf{P}} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T$.

$$\begin{align}\bar{\mathbf{x}} - \tilde{\mathbf{P}}\bar{\mathbf{x}} &= \bar{\mathbf{x}} - (\mathbf{P}\bar{\mathbf{x}} + \mathbf{u}\mathbf{u}^T\bar{\mathbf{x}}) \tag{4} \\ &= (\mathbf{I}_m - \mathbf{P})\bar{\mathbf{x}} - (\mathbf{I}_m - \mathbf{P})\mathbf{e}_1\mathbf{e}_1^T(\mathbf{I}_m - \mathbf{P})\bar{\mathbf{x}}/l^2 \tag{5} \\ &= (\mathbf{I}_m - \mathbf{P})\bar{\mathbf{x}} - (\mathbf{I}_m - \mathbf{P})\mathbf{e}_1\mathbf{e}_1^T\mathbf{r}/l^2 \tag{6} \\ &= (\mathbf{I}_m - \mathbf{P})\bar{\mathbf{x}} - (\mathbf{I}_m - \mathbf{P})[r_1,0,0,\cdots,0]^T/l^2 \notag\end{align}$$

(4) comes from the definition of $\tilde{\mathbf{P}}$, (5) from the definition of $\mathbf{u}$, and (6) from the definition of $\mathbf{r}$.

From (3) and (6), if $\hat{x}_1 = r_1/(P_{1,1}-1)$, then $\hat{\mathbf{x}} - \mathbf{P}\hat{\mathbf{x}} = \bar{\mathbf{x}} - \tilde{\mathbf{P}}\bar{\mathbf{x}}$ and $d(\hat{\mathbf{x}},\mathbf{U}) = d(\bar{\mathbf{x}},\hat{\mathbf{U}})$. **QED**

Note: In theory, if $\mathbf{e_1} \in span(\mathbf{U})$, $\hat{x}_1$ can be any value. However, it is not useful to assign an arbitrary value to x_1 in practice. So, not surprisingly, we regard x_1 as *unrecoverable* in this case.

Theorem 2 gives the explicit solution to a more general problem, where several entries are missing. Without loss of generality, assume the first k entries $\{x_i \mid i = 1,\cdots,k\}$ are unknown.

Theorem 2: If $span\{\mathbf{e}_i \mid i = 1,\cdots,k\} \bigcap span(\mathbf{U}) = \phi$, $[\hat{x}_1,\cdots,\hat{x}_k]^T = (\mathbf{P}_{1:k,1:k} - \mathbf{I}_k)^{-1}[r_1,\cdots,r_k]^T$ minimizes $d(\mathbf{x},\mathbf{U})$, where $\mathbf{P}$ is defined as in theorem 1, $\bar{\mathbf{x}} = [0,\cdots 0, x_{k+1}, x_{k+2},\cdots,x_m]^T$ and $\mathbf{r} = \bar{\mathbf{x}} - \mathbf{P}\bar{\mathbf{x}}$.

In theorem 2, if $span\{\mathbf{e}_i \mid i = 1,\cdots,k\} \bigcap span(\mathbf{U}) = \phi$, the unknown entries can not be recovered, as will be clear later in the discussion of the *recoverability*. The proof for theorem 2 is similar to theorem 1, except that the proof is much more complicated because of the introduction of a vector of unknown entries. The general idea is that we firstly prove the distances from all possible vectors to the subspace are less than or equal to a constant. Then, as done in theorem 1, we construct the vector, whose distance to the subspace is equal to that constant.

3.2 Recoverability of the unknown entries

Obviously, the recovery of the unknown entries depends on the existence of $(\mathbf{P}_{1:k,1:k} - \mathbf{I}_k)^{-1}$, from the proof above. Another criterion for the recoverability, as stated in the theorems, is $span\{\mathbf{e}_i \mid i = 1,\cdots,k\} \bigcap span(\mathbf{U}) = \phi$. Are there any relationships between them? How many unknown entries can be recovered at most? Theorem 3 gives a positive answer to the first question and theorem 4 provides a quantitative answer to the second.

Theorem 3: $span\{\mathbf{e}_i \mid i = 1,\cdots,k\} \bigcap span(\mathbf{U}) = \phi$ is equivalent to $\det(\mathbf{P}_{1:k,1:k} - \mathbf{I}_k) \neq 0$, i.e. $(\mathbf{P}_{1:k,1:k} - \mathbf{I}_k)^{-1}$ exists.

Proof: Because $\mathbf{U}$ is a rank-*r* orthonormal matrix, there exists another (always many) rank-$m-r$ orthonormal matrix $\mathbf{U}'$: which makes $[\mathbf{U},\mathbf{U}'] \in O^{m,m}$. We block the orthonormal matrix as: $[\mathbf{U},\mathbf{U}'] = \begin{bmatrix}\mathbf{S}_1\mathbf{S}_2 \\ \mathbf{S}_3\mathbf{S}_4\end{bmatrix}$, where $\mathbf{S}_1 \in R^{k,k}$, $\mathbf{S}_2$ and $\mathbf{S}_3^T \in R^{k,m-k}$, and $\mathbf{S}_4 \in R^{m-k,m-k}$. Then $\mathbf{P}_{1:k,1:k} = \mathbf{S}_1\mathbf{S}_1^T$, $\mathbf{S}_1\mathbf{S}_1^T + \mathbf{S}_2\mathbf{S}_2^T = \mathbf{I}_k$, and $\mathbf{P}_{1:k,1:k} - \mathbf{I}_k = -\mathbf{S}_2\mathbf{S}_2^T$. Similarly, $\mathbf{P} - \mathbf{I}_m = -\mathbf{U}'\mathbf{U}'^T$. So,

$$\begin{aligned}&\det(\mathbf{P}_{1:k,1:k} - \mathbf{I}_k) = 0 \\ &\Leftrightarrow \exists\ non-zero\ \mathbf{q} \in R^k, (\mathbf{P}_{1:k,1:k} - \mathbf{I}_k)\mathbf{q} = \mathbf{0} \\ &\Leftrightarrow \mathbf{q}^T(\mathbf{P}_{1:k,1:k} - \mathbf{I}_k)\mathbf{q} = \mathbf{0} \\ &\Leftrightarrow \begin{bmatrix}\mathbf{q} \\ \mathbf{0}_{m-k}\end{bmatrix}^T (\mathbf{P} - \mathbf{I}_m)\begin{bmatrix}\mathbf{q} \\ \mathbf{0}_{m-k}\end{bmatrix} = \mathbf{0} \\ &\Leftrightarrow (\mathbf{P} - \mathbf{I}_m)\begin{bmatrix}\mathbf{q} \\ \mathbf{0}_{m-k}\end{bmatrix} = \mathbf{0} \\ &\Leftrightarrow \begin{bmatrix}\mathbf{q} \\ \mathbf{0}_{m-k}\end{bmatrix} \in span(\mathbf{U})\end{aligned}$$

QED

Theorem 4: $m-r$ unknown entries can be recovered at most.

Proof First, we prove that $\mathbf{P}-\mathbf{I}_m$ has a rank of $m-r$.

$\mathbf{P}-\mathbf{I}_m = -\mathbf{U}'\mathbf{U}'^T$, where $\mathbf{U}'$ is defined in the proof of theorem 3. From the fact that $\mathbf{U}'$ is of rank $m-r$, $\mathbf{P}-\mathbf{I}_m$ has a rank of $m-r$. $\mathbf{P}_{1:k,1:k}-\mathbf{I}_k$, which is a submatrix of $\mathbf{P}-\mathbf{I}_m$, can have a $m-r$ rank at most. So, $m-r$ unknown entries can be recovered at most. **QED**

Note: The recoverability is obviously contingent on the number of the unknown entries. From theorem 4, if there are more than $m-r$ unknown entries, they are definitely beyond recovery. If not, from theorem 3, the recoverability can be determined in the recovering process, where the inverse of $\mathbf{P}_{1:k,1:k}-\mathbf{I}_k$ is needed. In practice, $\mathbf{P}_{1:k,1:k}-\mathbf{I}_k$ always has a rank of k, if $k \le m-r$, either in real-life data or in synthetic data, generated by computer. So, the missing data can always be recovered if the unknown number is less than or equal to $m-r$. However, the recovery is very sensitive to the noise if the unknown number equals to or is slightly less than $m-r$.

3.3 The convergence of the iterative algorithm

In this section, we prove the convergence of the algorithm, in section 2.2, in the simplest case, where only one column has some unknown elements. Without loss of generality, we suppose $\mathbf{M} = [\overline{\mathbf{M}}, \mathbf{m}]$, where $\mathbf{m}$ is the last column of $\mathbf{M}$ and has some unknown elements. The algorithm goes in this way.

Algorithm 2:

Initialization: Obtain the closest rank-*r* approximation, of $\overline{\mathbf{M}}$, by SVD: $\overline{\mathbf{M}}^r = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0$. From $\mathbf{U}_0$, obtain the estimate, $\hat{\mathbf{m}}_1$, of $\mathbf{m}$, by theorem 2. Set $\hat{\mathbf{M}}_1 = [\overline{\mathbf{M}}, \hat{\mathbf{m}}_1]$, and $d_0 = \infty$.

Iteration: From $\hat{\mathbf{M}}_i$, obtain its closest rank-*r* approximation by SVD: $\hat{\mathbf{M}}_i^r = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i$. Compute the distance $d_i = \| \hat{\mathbf{M}}_i^r - \hat{\mathbf{M}}_i \|_F$. If $d_{i-1} - d_i < \varepsilon$, terminate the iteration; else, from $\mathbf{U}_i$, recover $\mathbf{m}$ by theorem 2, and obtain $\hat{\mathbf{m}}_{i+1}$. Set $\hat{\mathbf{M}}_{i+1} = [\overline{\mathbf{M}}, \hat{\mathbf{m}}_{i+1}]$ and go to the iteration loop.

Theorem 5: The iterative algorithm above converges to a local minimum.

Proof

$$\begin{aligned} d^2(\hat{\mathbf{M}}_i, \mathbf{U}_i) &= d^2(\overline{\mathbf{M}}, \mathbf{U}_i) + d^2(\hat{\mathbf{m}}_i, \mathbf{U}_i) && (7) \\ &\ge d^2(\overline{\mathbf{M}}, \mathbf{U}_i) + d^2(\hat{\mathbf{m}}_{i+1}, \mathbf{U}_i) && (8) \\ &= d^2(\hat{\mathbf{M}}_{i+1}, \mathbf{U}_i) && (9) \\ &\ge d^2(\hat{\mathbf{M}}_{i+1}, \mathbf{U}_{i+1}) && (10) \end{aligned}$$

(7, 9) come from the definition of *d*, (8) from theorem 2, and (10) from the SVD theorem. **QED**

The proof of the convergence for **algorithm 1**, in section 2.2, is almost same as the proof above.

Note: on an alternative for the convergence condition

Another condition for the convergence, not so rigorous as (2) in the algorithm, is to check the variation of the unknown entries, i.e.

$$\| \hat{\mathbf{M}}_{i+1} - \hat{\mathbf{M}}_i \|_F < \varepsilon' \qquad (11)$$

Condition (11) is easier to check. ***However, it may sometimes happen that condition (11) fails to indicate convergence. We defer the explanation of the discrepancy between (2) and (11) until section 4.2.***

Note: global vs local minimums

In practice, in the problem of fitting a large matrix to its low-rank approximation, the known entries should provide enough information to recover the unknown ones. So, an approximate subspace can be obtained from the known entries and the initial recoveries (i.e. after the first stage of our algorithm) should be close to the optimal solution, as will be demonstrated in the experiments. Under this assumption, the iteration would almost certainly converge to the optimal solution, as observed in the experiments.

4 Experiments

In this section, we have two objectives: first, to better clarify the conditions for convergence and the global vs. local minimum behaviour. Our second objective is to evaluate the performance of the proposed approach in comparison with other approaches, especially with Jacobs'. We present 3 groups of experiments, two using synthetic data and one from the "box frames" sequence, which was also used by Jacobs [4,5].

4.1 Evaluation of the algorithm

Here, when evaluating the algorithm, we utilize a different index from that in Jacobs' approach [4,5]. When employing synthetic data, we evaluate the performance by the error between the rank-*r* closest approximation, $\hat{\mathbf{M}}^r$, of the recovered matrix and the noise-free matrix $\tilde{\mathbf{M}}$: $\| \hat{\mathbf{M}}^r - \tilde{\mathbf{M}} \|_F$. For real data, only a variant of this index can be used. Because noise-free data is not available, we use the error, $\| \hat{\mathbf{M}}^r - \mathbf{M}^r \|_F$, as the index for comparing the performance: $\mathbf{M}^r$ is the closest rank-*r* approximation of the real matrix $\mathbf{M}$.

4.2 Only one unknown entry

Consider a matrix $\tilde{\mathbf{M}} \in R^{10\times 10}$, whose rank is 3. $\tilde{\mathbf{M}}$ is corrupted with Gaussian noise (zero mean and unit variance) producing $\mathbf{M}$, which is observed. Specifically, in Matlab notation, $\widetilde{\mathbf{M}} = randn(10,3) \times randn(3,10) \times 5$ and $\mathbf{M} = \widetilde{\mathbf{M}} + randn(10,10)$. Suppose a single element, $M_{1,10}$, is unknown.

In this experiment, in order to evaluate the algorithm, we also search the neighborhood of the candidate solution, by perturbing the estimated value, $\hat{M}_{1,10}$. We compute the distances of 200 perturbed matrices, $\hat{\mathbf{M}}$, respectively to their rank-3 approximations, $\hat{\mathbf{M}}^3$, where $\hat{M}_{1,10}$ takes one of 200 values centred around $\hat{M}_{1,10}$, i.e., $\hat{M}^i_{1,10} = step \times i + \hat{M}_{1,10}$ for $i = -100:-1, \quad 1:100$. When the step is small (e.g., 0.1), we search a small area; while a large step (e.g., 3) is used to search a large area. Fig. 1 shows two of these experiments, one of which is denoted by the solid curves and the other by the dotted curves. Two curves in the lower part are from the experiment using a smaller step and the other two curves from the larger step. The horizontal axes are the step numbers in the above recipe for generating the perturbations: the point 0 is the solution obtained by the iterative algorithm. Note: thus the scales of the upper and lower graphs are different – the lower curves are in fact an expanded part of the upper curves. From the smaller steps, the solution appears to be a local minimum. From the larger step, we may see other local minimums or maximums.

Thus we can see examples of the iteration behaviour: suppose, for example, that the initial value of $M_{1,10}$ in the matrix corresponding to the solid-curve example is assigned the value $\hat{M}_{1,10} + 3 \times 80$, which is shown as the star, "*", on the solid curve. Starting from here, the algorithm can't find the correct solution. Even worse, when the convergence condition is criterion (11), the iterations will proceed to the infinite if there is no other local minimum in the right part, i.e., if the convergence condition is $\| \hat{\mathbf{M}}_{i+1} - \hat{\mathbf{M}}_i \|_F < \varepsilon'$, defined in (11), the algorithm will not converge. However, the iteration will stop under the condition of $d_{i-1} - d_i < \varepsilon$ - in effect, due to the extremely gentle slope of the curve, large changes in abscissa (related to (11)) produce only small changes in the ordinate (related to (2)). ***<u>Those cases, non-convergent measured by (11), are described non-convergent in sections 4.3 and 4.4.</u>***

We have run the experiments 1000 times, and in all of them we found good solutions, which can be regarded as the global minimum. First, the recovered data is closer to the noise-free data than the noise-corrupted one. Secondly, the distance of the noise-corrupted matrix to its rank-3 approximation is almost same as the solution by the algorithm. Thirdly, compared with the other 200 perturbed matrices selected in a large or small area around the solution produced by our method, that solution is the best one, as shown in fig. 1. It has to been admitted that such sampling strategy can never totally rule out the existence of other better solutions within the sampling area. However, the optimal solution, if it is not the one obtained by our approach, must lie beyond the large searched area because of the smoothness of the objective function (as observed in fig. 1) and it is not meaningful in practice. From the experiments in section 4.3, we can also draw such an experimental conclusion: the solution can be regarded as the optimal estimate, because $\| \hat{\mathbf{M}}^r - \widetilde{\mathbf{M}} \|_F$ is very close to $\| \mathbf{M}^r - \widetilde{\mathbf{M}} \|_F$, which means that the recovered matrix almost has same error as the whole noised matrix.

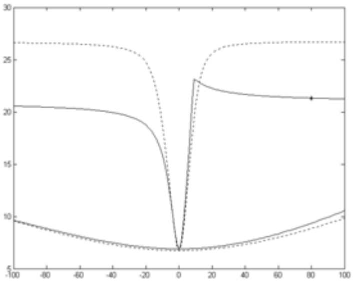

Fig. 1: Two 10-by-10 examples with one unknown entry

4.3 Synthetic data in a 40×40 matrix

In this subsection, we report the results of experiments on synthetic matrices with a rank of 3, which are then corrupted with Gaussian noise. Specifically, $\widetilde{\mathbf{M}} = randn(40,3) \times randn(3,40) \times 5$ and $\mathbf{M} = \widetilde{\mathbf{M}} + 0.1 \times randn(40,40)$. Because the proposed algorithm has to start from a complete sub-matrix, we suppose that the first 6×6 sub-matrix is always known and the unknown entries randomly distribute in the other part of the matrix. By comparing with Jacobs' method, we study, in the following, the performance of the proposed method at two stages: after the initialization step and after the iteration loop. Note: the error between $\mathbf{M}^3$ and $\widetilde{\mathbf{M}}$, $\| \mathbf{M}^3 - \widetilde{\mathbf{M}} \|_F$, is approximately ***1.5***.

(i) When the percentage of unknown entries is 10%, the proposed algorithm converges in all the 100 experiments. The performance of the iterative algorithm is very stable, indeed the error, $\| \hat{\mathbf{M}}^3 - \widetilde{\mathbf{M}} \|_F$, is always around ***1.6***; which is much smaller than the error for Jacobs' method, as shown in fig. 2. In all these 100 experiments, the initialization step of our approach also performs better than Jacobs' method.

(ii) When the percentage of the unknown entries goes up at 30%, all the cases still converge and its performance, always around ***1.9***, is still much better than Jacobs' method, as shown in fig. 3. In 76 out of the 100 cases, the initial estimation's performance is also better than Jacobs'.

(iii) When the percentage of the unknown entries goes up at 50%, 98 cases in all the experiments still converge and its performance, always around ***2.4***, is still much better than Jacobs' method, as shown in fig. 4. In the other two non-converged cases, Jacobs' method has errors of 17.9627 and 22.7680. In 68 out of the 98 cases, the initial estimation's performance is better than Jacobs'.

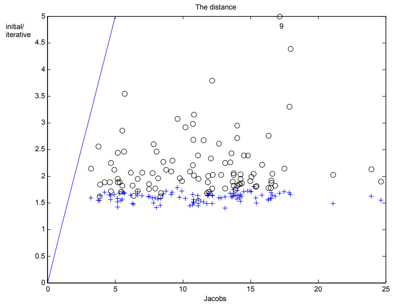

Fig. 2

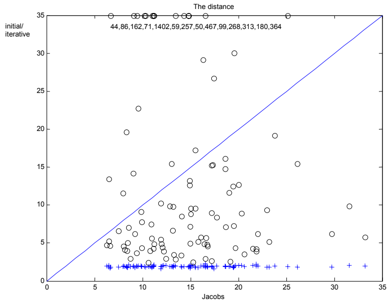

Fig. 3

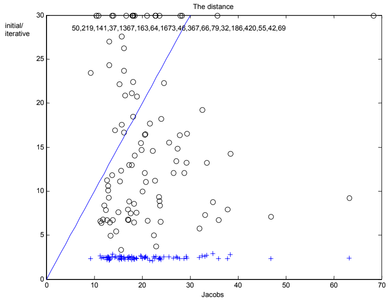

Fig. 4

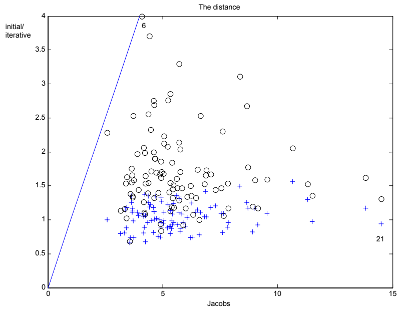

Fig. 5

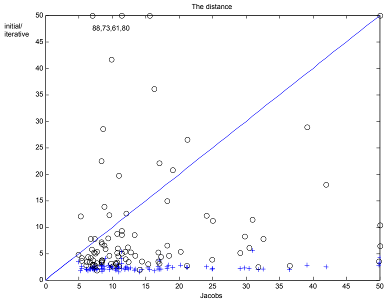

Fig. 6

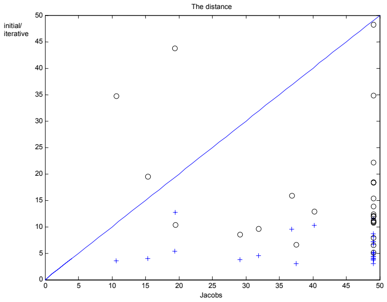

Fig. 7

Fig. 2-7: 40-by-40 matrix with 10% (fig.2), 30% (fig.3), and 50% (fig.4) of entries unknown, and box-frame with 10% (fig.5), 30% (fig.6), and 50% (fig.7) of entries unknown. In these figures, the horizontal axis is the index of the error by Jacobs' method and the vertical axis denotes the index for the initialization step and the iterative method. Symbol, "+", denotes the iterative method and "o" for the initialization step. So, a point, lying in the right to the x-y=0 line, means that Jacobs' method performs worse than the iterative method or the initialization step, vice versa.

4.4 Real-data experiment

Here, to test the algorithm on real data, we use the box video, which was used in [4,5]. The sequence consists of 40 feature points across 8 frames. One frame is shown in fig. 8. As in section 4.3, we suppose that 8 points in 4 frames are available. This 8×8 submatrix is randomly selected. We then randomly occlude the other feature points. In this experiment, we particularly note how often the recovery error, $\| \mathbf{M}^4 - \hat{\mathbf{M}}^4 \|_F$, is within a bound, like 20 or 50, because we find that the reconstruction error by the factorization method is probably unendurable when $\| \mathbf{M}^4 - \hat{\mathbf{M}}^4 \|_F > 50$, and the factorization method sometimes performs very bad even if $\| \mathbf{M}^4 - \hat{\mathbf{M}}^4 \|_F > 20$, in the box sequence. In addition, we do not, in this experiment, take into consideration the special nature of the problem of the factorization method in SFM: where "one of the vectors spanning its row space is known to be **1**, a vector of all ones" [5]. This task can be separately solved by the metric transformation [9,13] after recovering the missing data. Here, we take the recovery of the unknown entries as a general problem, stated in section 2.2, except that a point's x and y coordinates appear or disappear simultaneously.

(i) When the unknown entry level is about 10%, the performance measure from the iterative algorithm is around 1.1. The result is also good from the initialization step (alone) in all the cases. On 99 cases, Jacobs' method does not perform as well as the initialization step.

(ii) As the unknown entry level grows to 30%, 3 cases out of the 100 experiments can not be initialized because one or more columns (rows) have more than 12 (36) unknown entries. In another 2 cases, the iterative method does not converge, and the indexes for Jacobs' method are 10.0 and 8.4. In the remaining 94 cases, the performance measure is less than 4, for the iterative algorithm, and the mean is around 2.3. On 1 case, the error is about 5.6. In those cases that can be initialized, the errors in 4 cases are more than 50 for the initialization step, compared with 3 cases for Jacobs' method. On the whole, the initialization step performs better than Jacobs' method, as can be seen in Fig. 6.

(iii) When the unknown entry level grows up to 50%, 70 cases can not be initialized because of too may unknown entries in some rows (or some columns). In the other 30 cases, the iterative algorithm converges in 26 of them. Even if one takes Jacobs' result as the starting point, only 13 cases in all the 100 cases converge. In these 13 cases, algorithm 1 converges in 11 cases and fails in another 2 cases. While, there are other 15 cases where algorithm 1 converges and the iterative algorithm, taking Jacobs' result as the starting point, does not converge. We only list the 26 convergent cases in fig. 7, where Jacobs' method has more-than-50 error on 17 cases, shown near the right part. On the other 4 non-convergent cases, the indexes for Jacobs' method are 196.2, 33.6, 8258.8 and 16.2.

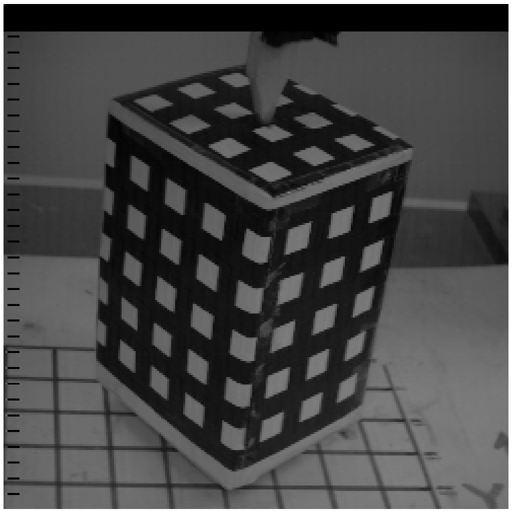

Fig. 8: One frame of the box sequence.

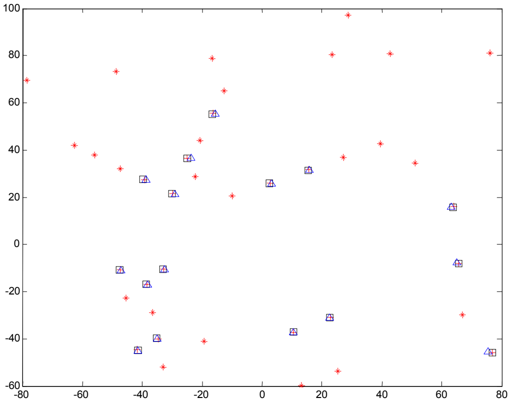

Fig. 9: The 5th frame in box video with 16 points occluded, where the unknown percentage in the whole sequence is 30%.

We give an example with about 30% of the unknown entries for the box video, in fig. 9, to see how well the algorithm recovers the missing data. Fig. 9 shows only one frame, the 5th frame, where are 24 points present in the video and other 16 points are artificially occluded. The 24 points, present in the video, are denoted by "*", and true positions of the occluded points are denoted by "+". The recovered positions by the iterative method are denoted by "□", and "Δ" for the initialization result. Three types of positions, for the occluded points, overlap in fig. 9, so that we even hardly see the difference between them.

5 Conclusion

The main contributions of this paper are to propose an iterative algorithm to the problem about missing data in a large low-rank matrix and to prove its convergence. The experiments show its effectiveness. Although the method has the disadvantage of theoretically finding a local minimum, the outcome can be regarded as global optimal solution if the noise level is not very strong or the percentage of missing entries is not very high; as verified by experiments with synthetic data and with real data. Another potential drawback of the approach, that having to start from a complete submatrix, will not present much difficulty in general.

References:

[1] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[2] M. Irani, Multi-frame optical flow estimation using subspace constraints, *ICCV99*(I)*,* pp. 626-633, 1999.

[3] M. Irani, Multi-frame correspondence estimation using subspace constraints, *IJCV*, vol. 48 (3), pp. 173-194, 2002

[4] D. Jacobs, Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images, *CVPR97,* pp. 206–212, 1997.

[5] D. Jacobs, Linear fitting with missing data for structure-from-motion, *CVIU*, vol. **82**, pp. 57–81, 2001.

[6] F. Kahl and A. Heyden, Affine structure and motion from points, lines and conics, *IJCV*, vol. **33**(3), pp. 163-180, 1999.

[7] K. Kanatani, Motion segmentation by subspace separation and model selection, *ICCV01*(II), pp. 301-306, 2001.

[8] D. Martinec and T. Pajdla, Structure from many perspective images with occlusion, *ECCV02*(II), pp.355-369, 2002.

[9] C. Poelman and T. Kanade, A paraperspective factorization method for shape and motion recovery, *IEEE PAMI*, vol. **19**(3), pp. 206–219, 1997.

[10] C. Rother and S. Carlsson, Linear multi view reconstruction with missing data, *ECCV02*(II), pp.309-324, 2002.

[11] H. Shum, K. Ikeuchi, and R. Reddy, Principal component analysis with missing data and its applications to polyhedral object modeling, *IEEE PAMI*, vol. **17**(9), pp. 854–867, 1995.

[12] P. Sturm and B. Triggs, A factorization based algorithm for multi-image projective structure and motion, *ECCV96*(II), pp.709-720, 1996.

[13] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: A factorization method, *IJCV*, vol. **9**(2), pp. 137–154, 1992.