

Robust structure from motion with affine camera via low-rank matrix recovery

WU Lun, WANG YongTian*, LIU Yue & WANG YuXi

*Key Laboratory of Photoelectronic Imaging Technology and System, Ministry of Education of China,
School of Optoelectronics, Beijing Institute of Technology, Beijing 100081, China*

Received September 16, 2012; accepted December 12, 2012; published online January 31, 2013

Abstract We present a novel approach to structure from motion that can deal with missing data and outliers with an affine camera. We model the corruptions as sparse error. Therefore the structure from motion problem is reduced to the problem of recovering a low-rank matrix from corrupted observations. We first decompose the matrix of trajectories of features into low-rank and sparse components by nuclear-norm and ℓ_1 -norm minimization, and then obtain the motion and structure from the low-rank components by the classical factorization method. Unlike previous methods, which have some drawbacks such as depending on the initial value selection and being sensitive to the large magnitude errors, our method uses a convex optimization technique that is guaranteed to recover the low-rank matrix from highly corrupted and incomplete observations. Experimental results demonstrate that the proposed approach is more efficient and robust to large-scale outliers.

Keywords robust PCA, structure from motion, low rank matrix, sparse errors

Citation Wu L, Wang Y T, Liu Y, et al. Robust structure from motion with affine camera via low-rank matrix recovery. *Sci China Inf Sci*, 2013, 56: 112102(10), doi: 10.1007/s11432-012-4728-y

1 Introduction

Structure from motion is a classical computer vision problem of finding the three-dimensional (3D) structure of scene and the camera motion. One family of batch structure from motion algorithms is called factorization methods which were originally proposed by Tomasi and Kanade [1]. Unlike sequential methods, factorization methods can simultaneously calculate camera motion and scene geometry using all image measurements. Gross errors associated with sequence closure can be avoided because reconstruction errors can be distributed meaningfully across all measurements.

The matrix of trajectories of features lies in a low-dimensional subspace. Therefore, the structure from motion problem can be reduced to that of finding the closest low-rank approximation to the matrix of trajectories. Conventional Principal Component Analysis (PCA) [2] is commonly used to obtain the best approximation of a low-rank structure in the least-squares sense, when the observation matrix is full, that is, when all the 2D track features can be observed in different frames. As we all know, PCA is guaranteed to be optimal solution only if the noise in measurement is independent and identically distributed (i.i.d.) Gaussian noise. Unfortunately, such a linear noise model rarely exists in reality: outliers and missing data are common factors that cause trouble in the structure from motion problem.

*Corresponding author(email: wyt@bit.edu.cn)

In order to remove outliers and find correct correspondence, a remarkable amount of information in the images is exploited, including analyzing neighboring structures in the image and some prior information. Kanade-Lucas-Tomasi feature tracker (KLT) [3] is commonly used in feature-tracking and feature-matching in very efficient and fast implementations when the two images are taken from close viewpoints. However, when features are tracked over an extended time period, this algorithm will fail because the estimation errors accumulate in time. Scale invariant feature transform (SIFT) [4] is invariant to changes in rotation, scale and illumination, and is more stable than KLT. Nevertheless, it still has problems when the two images are taken by cameras with wide baseline. Because of the presence of similarity between features, and other concurrent nuisance factors such as illumination changes or partial occlusions and corruptions, false matches cannot be totally avoided even with some robust statistics methods, such as RANdom SAMple Consensus (RANSAC) [5].

On the other hand, there are various techniques for the factorization method in the presence of missing data. Tomasi et al. [1] proposed an initialization method, which obtains the initial estimation by decomposing the largest full sub-matrix, and then grows it by adding one row or column at a time. However, finding the largest full sub-matrix is an non-deterministic polynomial-time hard (NP-hard) problem. Moreover, the solution largely relies on the order in which the columns or rows are completed. There is also a category of methods extended from Wiberg algorithms [6], such as those described in [7,8], which are all based on the Gauss-Newton method, and need to solve a large linear system whose cost function is interrelated to the column of the measurement matrix. The complexity will rapidly increase with the increase of the size of column. Jacobs proposed a linear fitting method [9], in which each column with missing data is treated as an affine subspace and unknown entries can be recovered by least-squares regression. But this algorithm is often inefficient due to the large noise in the data. Buchanan et al. [10] proposed the damped Newton (DN) algorithm based on the Levenberg-Marquardt algorithm. Although this method achieves a better performance than Wiberg's method, it needs a large number of iterations, which is not suitable for large matrices. Chen et al. [11] provided an iterative algorithm to recover the missing components in a low-rank matrix without large-scale noise. Starting with a complete sub-matrix, the algorithm fills the missing entries as matrix grows by one row or column. Later, Chen et al. [12] revitalized the usage of the Levenberg-Marquardt algorithm and redefined the objective function. The algorithm greatly reduces the number of variables and is much more efficient than the previous methods, but there are still some drawbacks such as requiring a good initial value and using the assumption that the data are corrupted by i.i.d. Gaussian noise.

All the above methods have some problems such as depending on the initial value selection and being sensitive to the large magnitude errors. These issues will cause the algorithm not converge or only converge to a local minimum. Motivated by the recent breakthroughs in low-rank matrix recovery [13–15], we present a novel approach to structure from motion that can simultaneously deal with missing data and outliers with an affine camera. We model the corruptions as sparse errors, thus the structure from motion problem is reduced to a problem of recovering a low-rank matrix from corrupted observations. Unlike previous methods, our method uses a convex optimization technique and enables robust recovery of a high-dimensional low-rank matrix from highly corrupted and incomplete observations. Experimental results demonstrate that the proposed approach is more efficient and robust to outliers. The remainder of this paper is organized as follows: In Section 2, we provide formulation associated with affine reconstruction. Section 3 gives the algorithm for solving the problem of recovering low-rank matrix from corrupted observation. Section 4 gives experimental results. Section 5 gives a summary of this paper.

2 Modeling structure from motion as matrix rank minimization

2.1 Affine projection

We first review the structure of trajectories matrix in a more general setting. In full generality, an affine camera has the form

$$P = \begin{pmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1)$$

After ignoring the third row of projection matrix, a homogeneous coordinates X_j^h of 3D point X_j and its projection on the camera imaging plane x_{ij} are related by a 2×4 affine projection matrix P_i :

$$x_{ij} = P_i X_j^h, \quad i = 1, \dots, m; \quad j = 1, \dots, n, \quad (2)$$

and then the full measurement matrix of rank 4 can be written as

$$O = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \begin{bmatrix} X_1^h & X_2^h & \dots & X_n^h \end{bmatrix} = \mathbf{P} \mathbf{X}^h. \quad (3)$$

2.2 Factorization method

We can rewrite Eq. (2) as

$$x_{ij} = \mathbf{M}_i \mathbf{X}_j + \mathbf{t}_i, \quad i = 1, \dots, m; \quad j = 1, \dots, n, \quad (4)$$

where \mathbf{M}_i represents the first third columns of projection matrix, \mathbf{t}_i represents translation vector, \mathbf{X}_j represents 3D point.

Without loss of generality, we assume that the origin is at the centroid of the 3D points. If matrix O in Eq. (3) is subtracted by its column-wise mean, we have

$$\begin{aligned} \hat{x}_{ij} &= x_{ij} - \frac{1}{n} \sum_{k=1}^n x_{ik} = \mathbf{M}_i \mathbf{X}_j + \mathbf{t}_i - \frac{1}{n} \sum_{k=1}^n (\mathbf{M}_i \mathbf{X}_k + \mathbf{t}_i) = \mathbf{M}_i \left(\mathbf{X}_j - \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \right) \\ &= \mathbf{M}_i \hat{\mathbf{X}}_j. \end{aligned} \quad (5)$$

We can write the full measurement matrix of size $2m \times n$ as Eq. (6), where \mathbf{M} and \mathbf{S} represent camera motion and object shape, respectively.

$$\hat{O} = \begin{pmatrix} \hat{x}_{11} & \hat{x}_{12} & \dots & \hat{x}_{1n} \\ \hat{x}_{21} & \hat{x}_{22} & \dots & \hat{x}_{2n} \\ \vdots & \vdots & & \vdots \\ \hat{x}_{m1} & \hat{x}_{m2} & \dots & \hat{x}_{mn} \end{pmatrix} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_m \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2 & \dots & \hat{\mathbf{X}}_n \end{bmatrix} = \mathbf{M} \mathbf{S}. \quad (6)$$

Once we have \hat{O} , pseudo motion $\tilde{\mathbf{M}}$ and pseudo shape $\tilde{\mathbf{S}}$ can be given by

$$\begin{cases} (\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = \text{svd}(\hat{O}), \\ \tilde{\mathbf{M}} = \mathbf{U} \mathbf{\Sigma}^{1/2}, \\ \tilde{\mathbf{S}} = \mathbf{V}^T \mathbf{\Sigma}^{1/2}. \end{cases} \quad (7)$$

The decomposition of Eq. (7) is not completely unique, there exists an unknown ambiguity $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ as follows:

$$\hat{O} = \tilde{\mathbf{M}} \mathbf{A} \mathbf{A}^{-1} \tilde{\mathbf{S}}, \quad (8)$$

this ambiguity can be reduced or removed with normalization constraints [16]. Then the true motion \mathbf{M} and shape \mathbf{S} can be obtained by

$$\mathbf{M} = \tilde{\mathbf{M}} \mathbf{A}, \quad \mathbf{S} = \mathbf{A}^{-1} \tilde{\mathbf{S}}. \quad (9)$$

However, when there exist corruptions, such as outliers and missing data, this solution normally breaks down. In the following sections, we will show that our method can efficiently remove these corruptions.

2.3 Formulation

As we discussed before, the low-rank structure of the observation matrix is seldom observed because of the corruptions in data. If we denote outliers by a matrix \mathbf{E} , the observation in Eq. (3) can be rewritten as

$$\mathbf{O} = \mathbf{P}\mathbf{X}^h + \mathbf{E}. \quad (10)$$

In most cases, the number of outliers is relatively small, i.e., only a few of entries in the \mathbf{E} are non-zero. Therefore, the structure from motion problem can be reduced to the optimization problem as

$$\min_{\mathbf{A}, \mathbf{E}} \text{rank}(\mathbf{A}) + \gamma \|\mathbf{E}\|_0 \quad \text{s.t.} \quad \mathbf{O} = \mathbf{A} + \mathbf{E}. \quad (11)$$

Because of the self-occlusion, missing data always occur in most of structure from motion applications. Taking this factor into account, the problem in Eq. (11) can be rewritten as

$$\min_{\mathbf{A}, \mathbf{E}} \text{rank}(\mathbf{A}) + \gamma \|\mathbf{E}\|_0 \quad \text{s.t.} \quad \pi_{\Omega^c}(\mathbf{O}) = \pi_{\Omega^c}(\mathbf{A} + \mathbf{E}), \quad (12)$$

where Ω^c denotes the locations of missing entries in the observed matrix \mathbf{O} . π_{Ω^c} denotes the orthogonal projection operator corresponding to the Ω^c , and Ω denotes the linear subspace complementary to Ω^c .

3 Efficient solution via convex optimization

The optimization problem in Eq. (11) is an NP-hard problem and not directly solvable. With the breakthroughs in low-rank matrix recovery, it was shown [13–15] that Eq. (11) can be solved by the strategy of convex surrogate, which replaces $\text{rank}(\cdot)$ with the nuclear norm and ℓ_0 -norm with matrix ℓ_1 -norm. Then, the problem is reduced to the optimization problem

$$\min_{\mathbf{A}, \mathbf{E}} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 \quad \text{s.t.} \quad \mathbf{O} = \mathbf{A} + \mathbf{E}, \quad (13)$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ represent the nuclear norm and ℓ_1 -norm, respectively, and $\lambda > 0$ is a weighting parameter. Then, our problem Eq. (12) can be rewritten as

$$\min_{\mathbf{A}, \mathbf{E}} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 \quad \text{s.t.} \quad \pi_{\Omega^c}(\mathbf{O}) = \pi_{\Omega^c}(\mathbf{A} + \mathbf{E}). \quad (14)$$

Recently, researchers have developed a lot of algorithms for solve low-rank matrix recovery [15,17,18]. Among them the augmented Lagrange multiplier(ALM) method [15,19] is faster and more stable than the others. Thus, we use ALM method [15] to solve our problem in Eq. (14), and the augmented Lagrangian is given by

$$\mathcal{L}_\mu(\mathbf{A}, \mathbf{E}, \mathbf{Y}) = \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 + \langle \mathbf{Y}, \pi_{\Omega^c}(\mathbf{O} - \mathbf{A} - \mathbf{E}) \rangle + \frac{\mu}{2} \|\pi_{\Omega^c}(\mathbf{O} - \mathbf{A} - \mathbf{E})\|_F^2, \quad (15)$$

where $\mathbf{Y} \in \mathbb{R}^{m \times n}$ is a Lagrange multiplier matrix, $\langle \cdot, \cdot \rangle$ represents the matrix inner product¹⁾, μ is a positive constant, and $\|\cdot\|_F$ is the Frobenius norm.

The basic ALM iteration is given by

$$\begin{cases} (\mathbf{A}_{k+1}, \mathbf{E}_{k+1}) = \text{argmin}_{\mathbf{A}, \mathbf{E}} \mathcal{L}_{\mu_k}(\mathbf{A}, \mathbf{E}, \mathbf{Y}_k), \\ \mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k \pi_{\Omega^c}(\mathbf{O} - \mathbf{A}_{k+1} - \mathbf{E}_{k+1}), \\ \mu_{k+1} = \rho \cdot \mu_k, \end{cases} \quad (16)$$

where $\{\mu_k\}$ is a monotonically increasing positive sequence ($\rho > 1$). \mathbf{A} and \mathbf{E} can be solved by an alternating minimization strategy as

$$\begin{cases} \mathbf{E}_{j+1} = \text{argmin}_{\mathbf{E}} \lambda \|\mathbf{E}\|_1 - \langle \mathbf{Y}_k, \pi_{\Omega^c}(\mathbf{E}) \rangle + \frac{\mu_k}{2} \|\pi_{\Omega^c}(\mathbf{O} - \mathbf{A}_j - \mathbf{E})\|_F^2, \\ \mathbf{A}_{j+1} = \text{argmin}_{\mathbf{A}} \|\mathbf{A}\|_* - \langle \mathbf{Y}_k, \pi_{\Omega^c}(\mathbf{A}) \rangle + \frac{\mu_k}{2} \|\pi_{\Omega^c}(\mathbf{O} - \mathbf{A} - \mathbf{E}_{j+1})\|_F^2. \end{cases} \quad (17)$$

1) $\langle \mathbf{X}, \mathbf{Y} \rangle \doteq \text{trace}(\mathbf{X}^T \mathbf{Y})$

For the first step in Eq. (17), there is a closed-form solution given by

$$E_{j+1} = \text{shrink} \left(\pi_{\Omega^c}(\mathbf{O}) + \frac{1}{\mu_k} \mathbf{Y}_k - \pi_{\Omega^c}(\mathbf{A}_j), \frac{\lambda}{\mu_k} \right), \quad (18)$$

where shrinkage is soft-thresholding operator and defined as follows:

$$\text{shrink}(x, \alpha) = \text{sign}(x) \cdot \max\{|x| - \alpha, 0\}, \quad (19)$$

where $\alpha > 0^2$.

Unfortunately, there is no close-form solution for the second step in Eq. (17). To overcome this problem, we use an iterative method so that the solution can converge reasonably. This method is based on the accelerated proximal gradient (APG) algorithm [17,18,20], which has better convergence and steadiness. The iterative solution is given as

$$\begin{cases} (\mathbf{U}_i, \boldsymbol{\Sigma}_i, \mathbf{V}_i) = \text{svd} \left(\frac{1}{\mu_k} \mathbf{Y}_k + \pi_{\Omega^c}(\mathbf{O}) - \mathbf{E}_{j+1} + \pi_{\Omega}(\mathbf{Z}_i) \right), \\ \mathbf{A}_{i+1} = \mathbf{U}_i \text{shrink} \left(\boldsymbol{\Sigma}_i, \frac{1}{\mu_k} \right) \mathbf{V}_i^T, \\ \mathbf{Z}_{i+1} = \mathbf{A}_{i+1} + \frac{t_i - 1}{t_{i+1}} (\mathbf{A}_{i+1} - \mathbf{A}_i), \end{cases} \quad (20)$$

where $\{t_i\}$ is a positive sequence and updated by $t_1 = 1$ and $t_{i+1} = 0.5 \left(1 + \sqrt{1 + 4t_i^2} \right)$ in each iteration. The entire algorithm is illustrated in Algorithm 1.

Algorithm 1 (Matrix completion and recovery via ALM).

Input: $\mathbf{O} \in \mathbb{R}^{m \times n}$, $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$, $\lambda > 0$.

Initialize $A_1 \leftarrow 0$, $E_1 \leftarrow 0$, $Y_1 \leftarrow 0$.

while not converged ($k = 1, 2, \dots$) **do**

$A_{k,1} = A_k$, $E_{k,1} = E_k$;

while not converged ($j = 1, 2, \dots$) **do**

$E_{k,j+1} = \text{shrink} \left(\pi_{\Omega^c}(\mathbf{O}) + \frac{1}{\mu_k} Y_k - \pi_{\Omega^c}(A_{k,j}), \frac{\lambda}{\mu_k} \right)$; $t_1 = 1$; $Z_1 = A_{k,j}$; $A_{k,j,1} = A_{k,j}$;

while not converged ($i = 1, 2, \dots$) **do**

$(\mathbf{U}_i, \boldsymbol{\Sigma}_i, \mathbf{V}_i) = \text{svd} \left(\frac{1}{\mu_k} Y_k + \pi_{\Omega^c}(\mathbf{O}) - E_{k,j+1} + \pi_{\Omega}(\mathbf{Z}_i) \right)$;

$A_{k,j,i+1} = \mathbf{U}_i \text{shrink} \left(\boldsymbol{\Sigma}_i, \frac{1}{\mu_k} \right) \mathbf{V}_i^T$, $t_{i+1} = 0.5 \left(1 + \sqrt{1 + 4t_i^2} \right)$;

$Z_{i+1} = A_{k,j,i+1} + \frac{t_i - 1}{t_{i+1}} (A_{k,j,i+1} - A_{k,j,i})$, $A_{k,j+1} = A_{k,j,i+1}$;

end while

$A_{k+1} = A_{k,j+1}$; $E_{k+1} = E_{k,j+1}$;

end while

$Y_{k+1} = Y_k + \mu_k \pi_{\Omega^c}(\mathbf{O} - A_{k+1} - E_{k+1})$, $\mu_{k+1} = \rho \cdot \mu_k$;

end while

Output: $(\hat{A}, \hat{E}) = (A_k, E_k)$.

4 Experiments

In this section, we present the experimental results of our method on synthetic and real data. We compare our results with those obtained by the Levenberg-Marquardt algorithm on Grassmann manifold (LMM) [12], which is formulated under the assumption of Gaussian noise. There are two steps for LMM method, namely initializing the system by alternated least squares (ALS) method and optimizing the system by Levenberg-Marquardt algorithm. The LMM is much more efficient than the other exist methods, therefore we use this method for comparison.

2) If $\alpha = 0$, then the shrinkage operator reduces to the identity operator

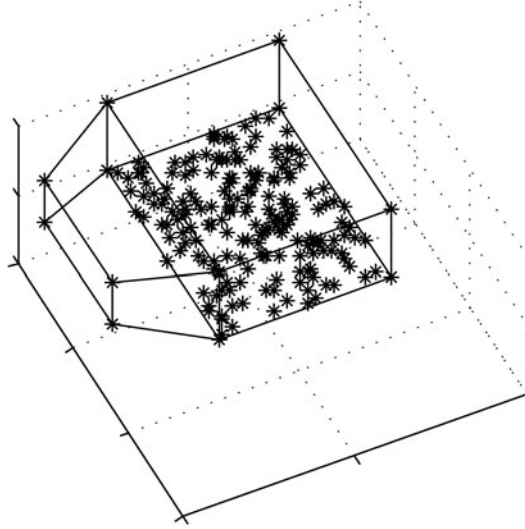


Figure 1 Synthetic data. Box with 200 features.



Figure 2 Trajectories. From left to right, Ground truth, recovered trajectories by our method, recovered trajectories by LMM.

4.1 Experimental results with synthetic data

In this subsection, we use synthetic data, for which the ground truth of trajectory matrix is known, to verify our method. We first create a set of points, and then create the camera motion around those points in a semi circle. A sparse noise with varying magnitudes is added to the trajectories matrix, which simulates outliers. We set $\lambda = 0.4/\sqrt{m}$ in Eq. (14).

a. With outliers. We generate a set of 200 3D points with a structure of box (see Figure 1). We then generate synthetic images with 60 cameras around those points, where the missing entries are chosen on the geometry relationship, and the locations of outliers are randomly chosen from the synthetic images. The two type of corruptions (missing entries and outliers) in the observation are with the percentages of 10% and 6%, respectively. The magnitude for outliers is randomly sampled in the range of 0–20.

The estimated trajectories are illustrated in Figure 2. We use the root-mean-square(RMS) error and the maximum projection error to evaluate both methods. The maximum projection error is defined as

$$\max(\|x_{ij}^g - x_{ij}^e\|_2), \quad i = 1, \dots, m; \quad j = 1, \dots, n, \quad (21)$$

where x_{ij}^g and x_{ij}^e are the ground truth and the estimated feature coordinate, respectively.

From Table 1 we can see that our method performs much better than the LMM approach. The errors almost occur in all features with LMM approach as shown in Figure 2. Since the data of box has a distinct box structure, we present its recovered structure in Figure 3, Figure 3 (a) shows the whole structure, Figure 3 (b) shows the structure in x - y plane. Note some lines are not parallel with LMM approach, while they are kept parallel with our method.

Table 1 A quantitative evaluation on data with corruptions

Error type	Algorithm	Error (in pixels)
Max projection error	LMM	8.2613
	Our method	0.0206
RMS error	LMM	1.1258
	Our Method	0.0005

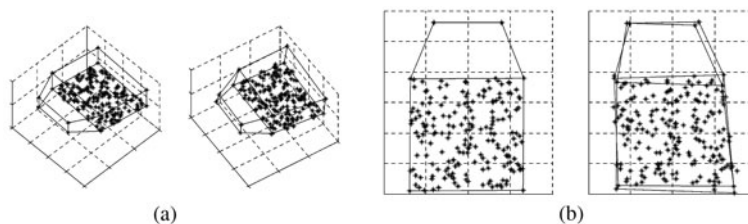


Figure 3 Recovered structure. (a) The whole structure. Left by our method, right by LMM; (b) the structure in x - y plane. Left by our method, right by LMM. Note: some lines are not parallel by LMM, while they are kept parallel by our method.

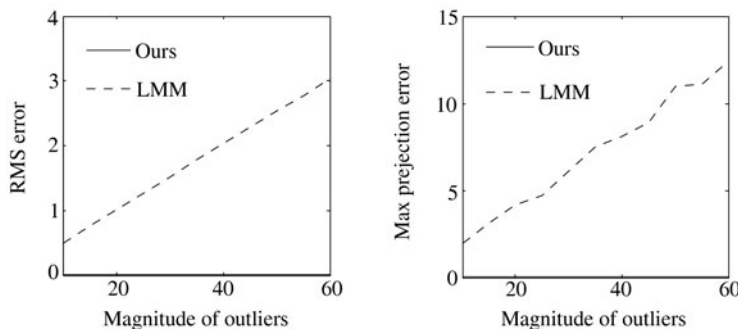


Figure 4 Effect of increasing magnitude of outliers. The locations of outliers are chosen at random from synthetic images. The results have been averaged over 10 different sets of locations for outliers.

b. Effect of magnitude of outliers. From the above experiments, we observe that the outliers have a powerful influence on the trajectory estimation. In this subsection, the effect of the magnitude of outliers is studied.

We vary the upper bound of magnitude of outlier from 10 to 60. The locations of outliers are randomly chosen from the synthetic data. We test our method on 10 different sets of locations for outliers. The average results are illustrated in Figure 4. We observe both the RMS error and the maximum projection error in the LMM estimate increase rapidly with the magnitude of outliers, while our method is significantly stable.

The experimental results demonstrate that our method is robust to the large magnitude of errors, whereas the LMM method is extremely sensitive to even small magnitude outliers in the input images. This is because our objective function uses l_1 -norm for errors, which is independent of the magnitude, while LMM approach uses l_2 -norm under the assumption of Gaussian noise, which is relevant to the magnitude of error.

c. Effect of amount of outliers. From the above experiments, we can see that our method is much more robust to outliers than the LMM method. In this experiment, we will test the effect of amount of outliers.

We generate data without missing entries, and then increase ratio of outliers. The locations of outliers are randomly chosen from synthetic images.

Again, we test our method on 10 different sets of locations for outliers. The average experimental results are shown in Figure 5. Our method can handle the data in which even 35% entries are corrupted

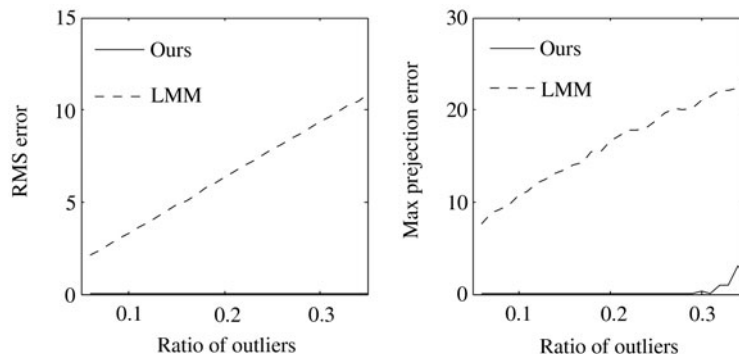


Figure 5 Effect of increasing ratio of outliers. The data are generated without missing entries, and the locations of outliers are randomly chosen from synthetic images.

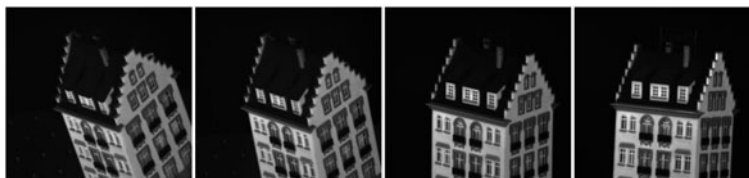


Figure 6 Sample images on real data.

by outliers, while the errors of the LMM methods rapidly increases when the percentage of noise increases. Again, our method performs much better than the LMM method.

d. Computation. We use SVD for each iteration of our algorithm, which has a complexity of $O(mn^2)$, where m and n represent the size of row and column, respectively. However, to speed up the algorithm, only partial SVDs are computed in each iteration since we expect the optimal solution to have a rank of 3 at most. Thus, the complexity of each iteration reduces to $O(lmn)$, where l is the number of singular vectors computed in each iteration. Typically, we set l to be much smaller than m and n . On the contrary, the LMM method has a complexity of $O(m^3r^3)$ for its main process, and its $O(nm^2r^2)$ for initialization step, where r is the rank of matrix. Our rank minimization approach can be used for structure from motion with much less computational cost than using the LMM approach.

For the data with Gaussian distribution in the first experiments, which consists of 200 3D points and 60 cameras with 10% missing entries and 6% outliers in observation matrix, on a Macintosh computer with 4 GB memory and a 2.66 GHz Core 2 i7 processor, our method takes about 5.1139 s, and the LMM method takes about 43.7833 s.

4.2 Qualitative evaluation with real images

In this experiments, we evaluate our method on real data. We use a set of 100 images of a house [21] taken under different views (see Figure 6). We use the KLT tracker [3] for feature tracking and intentionally relax the error threshold to test robustness of algorithms. We test on four sets of data, which contain 5.1%, 14.65%, 18.28%, 20.59% missing entries and correspond to the feature track 1–100, 100–199, 200–299 and 300–500, respectively. Since more features with similar descriptor are selected, the ratio of outliers is also increasing³⁾.

The experimental results are illustrated in Figures 7 (a)–(d), respectively. In each sub-figure, recovered trajectories are shown in the first row, recovered structures are shown in the second row, and top-down views of recovered structures are illustrated in the last row. Results by our method are on the left, and results by LMM are on the right.

It can be seen from Figure 7 (a) and (b) that both methods work well when ratio of corruptions are small. For the LMM method, errors are increasing gradually, as the ratio of corruption is raised. From

³⁾We cannot give exact ratio of outlier since no ground truth is available for outliers

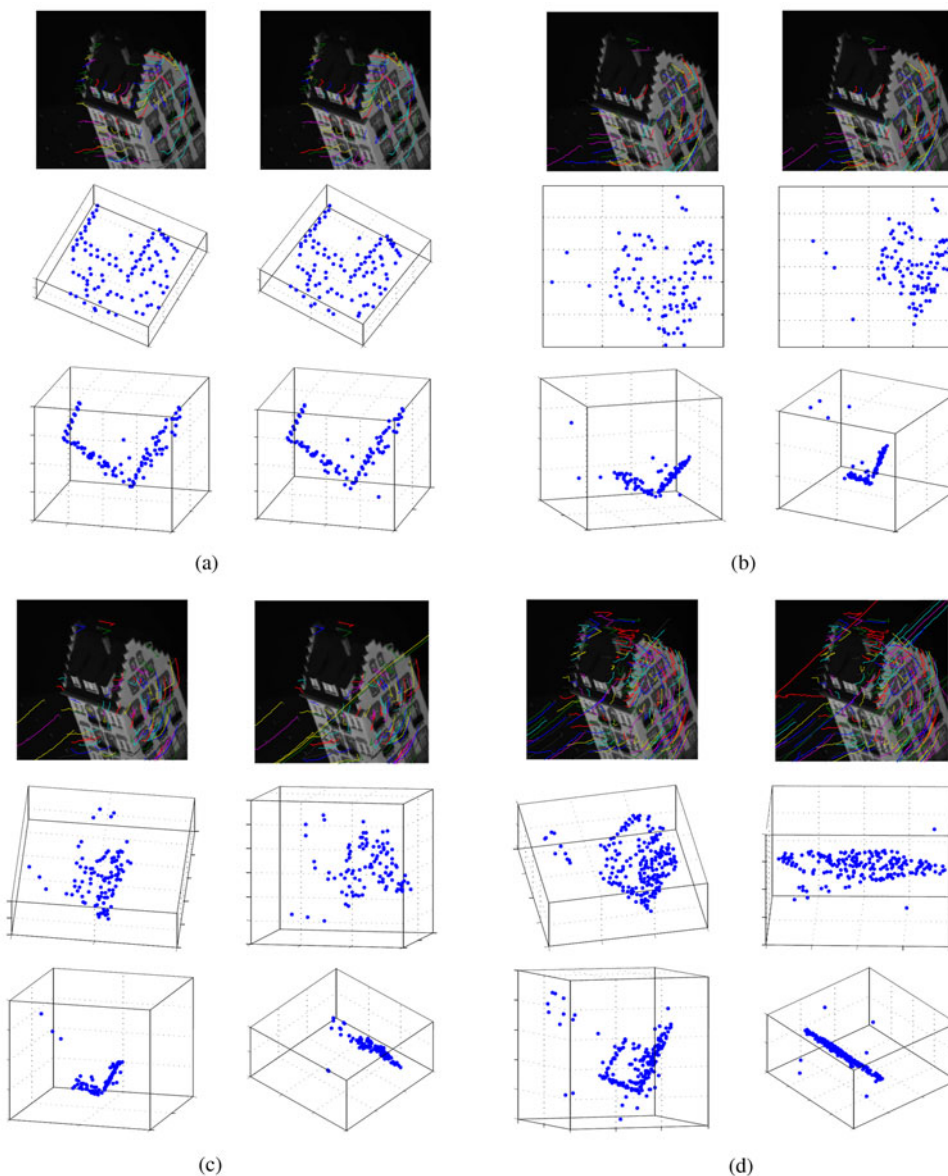


Figure 7 Real data. (a) (b) Recovered structure with low corruption: left by our method, and right by the LMM method; (c)(d) recovered structure with high corruption. left by our method, and right by the LMM method. Recovered trajectories are shown in the first row, recovered structures are shown in the second row, top-down views of recovered structure are shown in the last row.

Figure 7 (c) and (d), we can observe the distinct incorrect trajectories recovered by the LMM method, while the trajectories recovered by our method are more reasonable. From the top-down view, we can also observe that the angle between the two walls recovered by our method keeps perpendicular, while the structure recovered by the LMM method is degenerated to a plane. Our method shows higher robustness to errors.

5 Discussion and future work

In this paper, we present a novel approach to structure from motion that can deal with missing data and outliers simultaneously with an affine camera. The corruptions in data are modeled as sparse errors, and thus the structure from motion problem is reduced to the problem of recovering the low-rank matrix from corrupted observations.

Since our formulation is based on advanced convex optimization for nuclear norm and ℓ_1 -norm minimization, it does not rely on the initial value. Unlike previous methods, which have some drawbacks such as requiring a good initialization and being sensitive to the large magnitude outliers, our method uses a convex optimization technique that is guaranteed to correctly recover low-rank matrix from highly corrupted and incomplete observations. However, this framework is still limited to the affine camera model. We will extend our method to the perspective camera model in the future.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 60827003, 61072096)

References

- 1 Tomasi C, Kanade T. Shape and motion from image streams under orthography: a factorization method. *Int J Comput Vision*, 1992, 9: 137–154
- 2 Jolliffe I. *Principal Component Analysis*. Berlin: Springer-Verlag, 1986.
- 3 Tomasi C, Kanade T. Detection and tracking of point features. Technical Report. 1991
- 4 Lowe G D. Distinctive image features from scale-invariant keypoints. *Int J Comput Vision*, 2004, 60: 91–110
- 5 Fischler M, Bolles R. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. Chapter Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. San Francisco: Morgan Kaufmann Publishers Inc, 1987
- 6 Wiberg T. Computation of principal components when data are missing. In: *Proceedings of 2nd Symposium on Computational Statistics*, Berlin, 1976. 229–236
- 7 Okatani T, Deguchi K. On the wiberg algorithm for matrix factorization in the presence of missing components. *Int J Comput Vision*, 2007, 72: 329–337
- 8 Eriksson A, Anton Van D H. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm. In: *Proceedings of Computer Vision and Pattern Recognition*. San Francisco: IEEE Computer Society, 2010. 771–778
- 9 Jacobs D. Linear fitting with missing data: applications to structure-from-motion and to characterizing intensity images. In: *Proceedings of Computer Vision and Pattern Recognition*. Washington: IEEE Computer Society, 1997. 206–212
- 10 Buchanan A, Fitzgibbon A. Damped Newton algorithms for matrix factorization with missing data. In: *Proceedings of Computer Vision and Pattern Recognition*. Washington: IEEE Computer Society, 2005. 316–322
- 11 Chen P, Suter D. Recovering the missing components in a large noisy low-rank matrix: application to sfm. In: *Proceedings of Computer Vision and Pattern Recognition*. Washington: IEEE Computer Society, 2004. 1051–1063
- 12 Chen P, Suter D. Optimization algorithms on subspaces: revisiting missing data problem in low-rank matrix. *Int J Comput Vision*, 2008, 80: 125–142
- 13 Wright J, Ganesh A, Rao S, et al. Robust principal component analysis: exact recovery of corrupted low-rank matrices by convex optimization. In: *Proceedings of Neural Information Processing Systems*, 2009
- 14 Chandrasekaran V, Sanghavi S, Parrilo P, et al. Sparse and low-rank matrix decompositions. In: *Proceedings of IFAC Symposium on System Identification*. Piscataway: IEEE Press, 2009. 962–967
- 15 Lin Z, Chen M, Wu L, et al. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC Technical Report UILU-ENG-09-2215. 2009
- 16 Morita T, Kanade T. A sequential factorization method for recovering shape and motion from image streams. *IEEE Trans Pattern Anal*, 1997, 19: 858–867
- 17 Ganesh A, Lin Z, Wright J, et al. Fast algorithms for recovering a corrupted low-rank matrix. In: *Proceedings of Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. Aruba: IEEE Computer Society, 2009. 213–216
- 18 Toh K, Yun S. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pac J Optim*, 2006, 6: 615–640
- 19 Bertsekas D. *Nonlinear Programming*. 2nd ed. Cambridge: Athena Scientific, 2004
- 20 Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problem. *SIAM J Imag Sci*, 2009, 2: 183–202
- 21 CMU image data base: hotel. <http://vasc.ri.cmu.edu/idb/html/motion/hotel/index.html>