Factorization for Projective and Metric Reconstruction via Truncated Nuclear Norm

Yang Lin*†, Li Yang*†, Zhouchen Lin*†✉, Tong Lin*†, and Hongbin Zha*†
*Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, P. R. China
†Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, P. R. China.
Email: linyang@cis.pku.edu.cn, yangli@cis.pku.edu.cn, zlin@pku.edu.cn, lintong@pku.edu.cn, zha@cis.pku.edu.cn

***Abstract*—Structure from motion (SfM) is a crucial and widely studied problem in computer vision. Recently, the factorization framework for SfM was formulated as a low rank approximation problem: the rank of rescaled measurement matrix is always smaller than four. Since the rank function is non-convex, a common practice is to replace with its convex surrogate, i.e., the nuclear norm. However, nuclear norm sometimes gets unsatisfactory results. In this paper, we apply the recently proposed *truncated nuclear norm* to handle the factorization framework in a non-convex way, which heavily penalizes the singular values beyond the desired rank. We further introduce weighted ℓ_1-norm to handle missing data and outliers uniformly. Based on truncated nuclear norm, we propose two factorization models for projective reconstruction and metric reconstruction, respectively. We also proposed an extremely efficient algorithm to tackle one of the optimization sub-problems. Extensive experiments on synthetic and real datasets verify the effectiveness of our method for projective and metric reconstructions. Our method achieves higher accuracy in 3D reconstruction and is more robust to missing data and outliers.**

I. Introduction

The structure from motion (SfM) problem, i.e., recovering the 3D scene structure and camera motion from point correspondence, is an important issue in computer vision [1], [2]. Traditional methods tackle the problem via the epipolar, trifocal, or more constraints. However, these kinds of multilinear constraints cannot estimate the SfM parameters with more than four views simultaneously. In contrast, by adopting factorization framework, SfM can be solved uniformly.

Figure 1 illustrates a common framework of factorization methods for 3D reconstruction from multiple uncalibrated images. Given an uncalibrated image sequence of a scene, the feature matching and tracking module can be described by a measurement matrix $\mathbf{M}$, whose elements are the coordinates of matched features in the image sequence. The projective reconstruction module aims at recovering the projective depth matrix $\mathbf{\Lambda}$ or the corresponding rescaled measurement matrix (RMM) $\mathbf{W} \equiv \mathbf{\Lambda} \odot \mathbf{M}$ (where $\odot$ denotes the Hadamard product) and further factorizing $\mathbf{W}$ into a camera projection matrix $\widehat{\mathbf{P}}$ and a 3D structure matrix $\widehat{\mathbf{X}}$. The projective reconstruction differs from the true reconstruction by a projective transformation. Although projective factorization can provide some useful information, in most cases, it is the metric reconstruction that really matters, which only differs from the true reconstruction by a similarity transformation

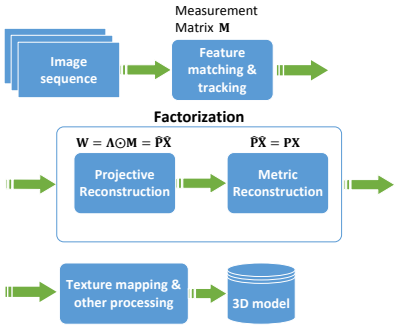


Fig. 1. A framework of factorization methods for 3D reconstruction. The two steps in the middle are the focus issues of this paper.

(composed of a translation, a rotation or a reflection, and a uniform change of scale). By assuming additional constraints, the projective reconstruction $\widehat{\mathbf{P}}\widehat{\mathbf{X}}$ can be upgraded to a metric reconstruction $\mathbf{PX}$. With texture mapping and other processing, we can build and visualize a 3D model. In this paper, we focus on the projection and metric reconstructions and propose new factorization methods for them.

A. Related work

The factorization method is originated from Tomasi and Kanade's work [3] based on an orthographic imaging model, where the projective depths can be assumed to be constant and hence the measurement matrix $\mathbf{M}$ can be directly factorized into $\mathbf{P}$ and $\mathbf{X}$ by singular value decomposition (SVD). A more general but also more difficult case is the perspective imaging model with unknown projective depths. The most popular approach is proposed by Sturm and Triggs [4], where the rescaled measurement matrix $\mathbf{W} = \mathbf{\Lambda} \odot \mathbf{M}$ is factorized into $\mathbf{W} = \hat{\mathbf{P}}\hat{\mathbf{X}}$, by alternately updating $\hat{\mathbf{P}}$, $\hat{\mathbf{X}}$, and $\mathbf{\Lambda}$. Despite its elegance, the Sturm-Triggs factorization and its extensions [5] have some disadvantages [1]: (1) the iteration is not guaranteed to converge; (2) sometimes it converges to a trivial solution; (3) the iteration may be unstable; (4) it works only when the perspective projection is close to be affine.

Recently, Dai et al. proposed an element-wise factorization (EWF) algorithm [6] that recovers $\mathbf{W}$ by solving a rank minimization problem. Angst et al. [7] further proposed an extension of EWF with the prior that the camera track is *smooth*. Since the rank minimization is NP-hard, the rank function is usually relaxed to its convex surrogate, i.e. the nuclear norm. The nuclear norm (also known as the trace

norm) of a matrix is defined as the sum of its singular values. However, the nuclear norm minimization cannot guarantee that the optimal $\mathbf{W}$ satisfies the fixed rank constraint.

Another category of low-rank matrix approximation is based on matrix factorization. One of the widely used matrix factorization methods is the so called Wiberg method [8], which jointly optimizes a product $\mathbf{U}\mathbf{V}^T$ of two fixed size $\mathbf{U}$ and $\mathbf{V}$ matrices, and introduces ℓ_1-norm to reject outliers. However, for large size and high-dimensional data, Wiberg method costs too much and requires more memory to reach the minimum. Compared with nuclear norm minimization, matrix factorization based methods can easily enforce the fixed rank constraint by the explicit bilinear matrix form, but introduces non-convexity, which makes the quality of the result dependent on initialization.

The projective reconstruction can convey some useful information, but the metric reconstruction is more attractable, because it is closer to the true reconstruction. There are two ways to achieve metric reconstruction:

(1) One is the classical approach called self-calibration, which first recovers the camera motion and parameters, and then computes the scene structure by triangulation. Most of the self-calibration methods are based on the Kruppa equations [9]. Mendonca and Cipolla [10] proposed a cost function minimization approach to recover the camera intrinsic parameters, which enjoys excellent convergence properties.

(2) The other is stratified reconstruction that upgrades projective reconstruction to metric reconstruction by exploiting many geometric constraints [11]. In most situations, these methods show a degradation as noise increases, and commonly require a initialization closed to the ground truth.

B. Contributions

While Dai et al. [6] and Angst et al. [7] formulate projective reconstruction as rank minimization problems, it is actually a *fixed rank* problem: the rank of the target matrix is known priori. Despite the nuclear norm has proven to be effective for solving rank minimization problems, it is deficient in tackling fixed rank problems. Therefore, we propose a novel method, which has following contributions:

- We apply truncated nuclear norm [12] to solve the fixed rank minimization problem. The truncated nuclear norm can encourage the rank of the optimal solution to be the desired one by heavily penalizes the singular values beyond the desired rank.
- We propose new models for both projective and metric reconstruction based on truncated nuclear norm, which greatly improve the accuracy of reconstructions. To enhance the robustness of reconstructions, we also introduce weighted ℓ_1-norm to handle outliers and missing data in a unified way. Moreover, our method works for both *uncalibrated* and *unstructured* camera motion, while [7] assumes that the camera path is *smooth*.
- We design an extremely efficient algorithm to tackle one of the optimization sub-problems and get better reconstruction result.

II. The proposed factorization methods

Our factorization methods can be divided into two steps (Figure 1). First, we recover the RMM and apply projective factorization. Second, we upgrade the projective reconstruction to the metric reconstruction. Both steps rely on truncated nuclear norm minimization.

A. Truncated nuclear norm

In SfM and many other computer vision problems, we are often faced with seeking an optimal matrix whose rank is known to be less than or equal to a given integer. Such problems are often hard to solve. A common practice is to find an intermediate solution by disregarding the rank constraint first, and then perform SVD to truncate the singular values beyond the given rank. Such a strategy is efficient in computation, but often the accuracy cannot be guaranteed. Alternatively, Dai et al. [6] proposed to solve a rank minimization problem for projective factorization. Since rank minimization problems are usually NP-hard [13], nuclear norm is used as a convex surrogate of the rank [6], [7], [13], [14], [15], [16]. Such a substitution has proven to be very successful in many low rank recovery problems. However, we observe that nuclear norm minimization is no longer effective for SfM problems.

For nuclear norm of matrices, the extreme points of unit nuclear norm ball are all rank-one matrices $\mathbf{u}\mathbf{v}^T$, where $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$. Suppose that the ambient space is consist of $p \times q$ matrices. Then the k-faces of the ball mainly consist of matrices whose ranks are $\min(k, p, q)$, since they are the convex combination of k rank-one $p \times q$ matrices. So we can infer that if the constraint subspace is r dimensional ($r < pq$), then for a linearly constrained nuclear norm minimization problem, the optimal solution is most likely to be rank $\min(pq - r, p, q)$. Therefore, if the dimension of the constraint subspace is very low, we cannot produce a truly low rank solution by using nuclear norm as a surrogate of rank. For our RMM recovery problem (Eq. (4)), we can easily count that the dimensions of the ambient space and the constraint subspace are $3m \times n$ and $mn - 1$, respectively. So if nuclear norm is used, we will most likely have a full rank solution, and 3D reconstruction errors will be large.

Notice that nuclear norm does not encode any prior knowledge about the rank constraint. It simply assigns an equal weight for all the singular values of the solution. However, for many SfM problems, we know that the rank of a solution should not exceed a fixed integer. To this end, in this paper we apply the recently proposed truncated nuclear norm [12] to get a more accurate rank minimization solution. The truncated nuclear norm is defined as the sum of $\min(m, n) - r$ minimum singular values, i.e.,

$$\|\mathbf{S}\|_{*,r} \equiv \sum_{i=r+1}^{\min(p,q)} \sigma_i(\mathbf{S}), \tag{1}$$

where $p \times q$ is the dimension of $\mathbf{S}$, $\{\sigma_i(\mathbf{S})\}$ are the singular values of $\mathbf{S}$, r is the known rank that $\mathbf{S}$ should not exceed.

B. Projective factorization

Basic model for recovering the rescaled measurement matrix. Suppose that there are n 3D points observed by m projective cameras. The image formation process can be described by $\lambda_{ij}\mathbf{m}_{ij} = \mathbf{P}_i\mathbf{x}_j$, where $\mathbf{P}_i \in \mathbb{R}^{3\times 4}$ is the projection matrix of the i-th camera, $\mathbf{x}_j = [x_j, y_j, z_j, 1]^T \in \mathbb{R}^4$ denotes the homogeneous coordinate of the j-th 3D point, $\mathbf{m}_{ij} = (u_{ij}, v_{ij}, 1)^T \in \mathbb{R}^3$ is the homogeneous coordinate of $\mathbf{x}_j$ in the image captured by the i-th camera, and λ_{ij} is a scaling factor, called the projective depth [1], to account for the perspective projection. Putting all imaged points together, we have a basic matrix equation [1], [2], [6]:

$$\mathbf{\Lambda} \odot \mathbf{M} = \mathbf{P}\mathbf{X}, \tag{2}$$

where $\mathbf{\Lambda} = [\lambda_{ij}] \otimes \mathbf{1}_{3\times 1} \in \mathbb{R}^{3m\times n}$, $\mathbf{M} = [\mathbf{m}_{ij}] \in \mathbb{R}^{3m\times n}$, $\mathbf{P} = [\mathbf{P}_1; \cdots; \mathbf{P}_m] \in \mathbb{R}^{3m\times 4}$, and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{4\times n}$. Here are some notations: $\odot$ represents the element-wise (Hadamard) product, $\otimes$ is the Kronecker product, and $\mathbf{1}$ is an all-one vector. $\mathbf{M}$ is called the measurement matrix. For simplicity, we define the rescaled measurement matrix as $\mathbf{W} \equiv \mathbf{\Lambda} \odot \mathbf{M} \in \mathbb{R}^{3m\times n}$.

To find the constraints on $\mathbf{W}$, we first observe that $\mathbf{\Lambda}$ can be replaced by $\lambda_{ij} = w_{3i,j}$ due to using homogeneous coordinates $\mathbf{m}_{ij} = (u_{ij}, v_{ij}, 1)^T$. So $\mathbf{W} - \mathbf{\Lambda} \odot \mathbf{M} = \mathbf{0}$ can be written as $\mathcal{A}(\mathbf{W}) = \mathbf{0}$, where

$$\mathcal{A}(\mathbf{W}) \equiv \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & w_{3i-2,j} - u_{ij}w_{3i,j} & \cdots \\ \cdots & w_{3i-1,j} - v_{ij}w_{3i,j} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}_{2m\times n}. \tag{3}$$

Moreover, it is easy to see that $\text{rank}(\mathbf{W}) \leq 4$, because it is expected that $\mathbf{W} = \mathbf{P}\mathbf{X}$ ($\mathbf{P} \in \mathbb{R}^{3m\times 4}$ and $\mathbf{X} \in \mathbb{R}^{4\times n}$). So the problem of finding $\mathbf{W}$ can be formulated as a fixed rank problem. As discussed in the last subsection, we use truncated nuclear norm [12] to enforce the rank constraint, resulting in the following optimization problem:

$$\min_{\mathbf{W}} \|\mathbf{W}\|_{*,r_1}, \quad \text{s.t.} \quad \mathcal{A}(\mathbf{W}) = \mathbf{0},\ \mathbf{d}^T\mathbf{W}\mathbf{1} = mn, \tag{4}$$

where $r_1 = 4$, and $\mathbf{d} = \mathbf{1}_{m\times 1}^T \otimes (0, 0, 1)^T$. The last constraint imposes $\sum_{ij}\lambda_{ij} = mn$ to avoid the trivial solution $\mathbf{W} = \mathbf{0}$. Note that the two constraints above are more general than those in [6]. Besides, we do not require λ_{ij} to be positive, allowing points to be behind some cameras, e.g., when taking photos on a moving car.

Dealing with outliers and missing data. In real SfM applications, outliers and missing data pose a great challenge to most factorization methods. Outliers may be caused by wrong matches. They make a small number of entries of $\mathbf{M}$ contaminated by gross errors, but in general we do not know where they are. For missing data, we know the exact position of these incomplete entries. They may be caused by occlusion or points out of image boundary or behind cameras. For both outliers and missing data, we assume that only a few entries are corrupted, thus being "sparse". Although this sparsity assumption seems a bit restrictive, we will show in Section IV that our method can also account for a reasonably high percentage of corruptions. Actually, we are able to fill in the measurement matrix even if it is very sparse (see Algorithm 1).

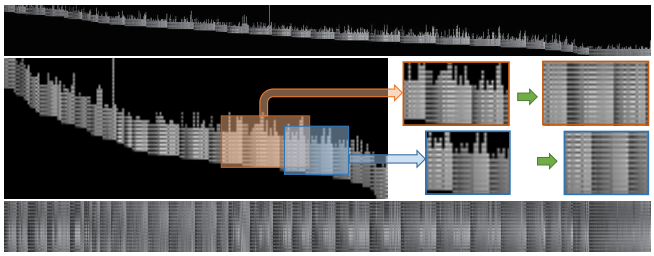


Fig. 2. Procedure to recover a highly sparse measurement matrix with a strong band-diagonal pattern. The black areas represent the missing data. Top row: the original measurement matrix. Second row: two sub-matrices with relatively few missing entries are selected from the original measurement matrix, then all the missing entries are filled in. Bottom row: the fully recovered measurement matrix. (**Figure in this paper are best viewed on screen!**)

In reality, outliers and missing data often mix together. To handle outliers and missing data jointly, we propose weighted ℓ_1-norm for these two types of corruptions, leading to the following model:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{E}} \quad & \|\mathbf{W}\|_{*,r_1} + \hat{\tau}\|\mathbf{E}\|_{\Omega\odot 1}, \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{W}) + \mathbf{E} = \mathbf{0},\ \mathbf{d}^T\mathbf{W}\mathbf{1} = mn, \end{aligned} \tag{5}$$

where $\mathbf{E}$ accounts for the corruptions, $\Omega \in \mathbb{R}^{2m\times n}$ is a 0-1 mask matrix:

$$\Omega = [\omega_{ij}],\ \omega_{ij} = \begin{cases} \mathbf{0} \in \mathbb{R}^2, & \text{if } \mathbf{m}_{ij} \text{ is missing,} \\ \mathbf{1} \in \mathbb{R}^2, & \text{if } \mathbf{m}_{ij} \text{ is available,} \end{cases} \tag{6}$$

and $\|\mathbf{E}\|_{\Omega\odot 1} \equiv \|\Omega \odot \mathbf{E}\|_1 = \sum_{ij}\omega_{ij}|e_{ij}|$, which only collects the errors at available pixels. In practice, we set $\hat{\tau} = \tau/\max\{3m, n\}$, where $\tau > 0$ is a tunable parameter. Eq. (5) can be efficiently solved by the alternating direction method of multiplier (ADMM) [17]. The details are given in Section III. We want to highlight that, if we mark those highly confident pixels as "available" pixels and those lowly confident ones as "missing" pixels, outliers can also be handled by Eq. (5).

Dealing with a highly sparse measurement matrix. According to [7], in real SfM applications, due to the occlusions, the measurement matrix $\mathbf{M}$ may be highly sparse, e.g., exhibiting a strong band-diagonal pattern with most of the off-band-diagonal entries missing (first row in Figure 2). Such a pattern is deemed to be very challenging for 3D recovery [7].

To tackle this issue, we propose an algorithm to iteratively fill in the missing entries with band-diagonal pattern. We first select multiple sub-matrices from $\mathbf{M}$ such that the percentage of the missing entries is smaller than a given threshold (second row in Figure 2). These sub-matrices are recovered in parallel using the above RMM recovery method. For the missing entries covered by multiple sub-matrices (Figure 3),

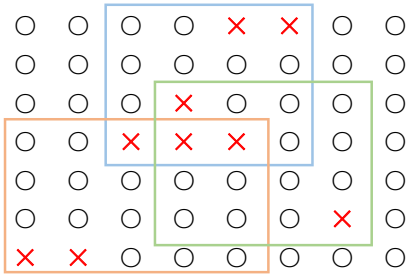


Fig. 3. A sketch of storing candidate values for missing entries of a measurement matrix. "×" represents a missing entry and "◯" represents a known entry. Three different rectangles represent three different $\widehat{\mathbf{M}}_{sub}$. It can be seen that some missing entries are covered by different $\widehat{\mathbf{M}}_{sub}$'s. So more than one candidate values are stored for this entry.

if the recovered values do not vary too much, their median values are assigned. Then we find new sub-matrices and fill in the missing entries again, until all the missing entries are filled in. This process is illustrated in Figure 2 and more details are given in Algorithm 1.

Algorithm 1 Process of Recovering a Highly Sparse Measurement Matrix

Input:

Original highly sparse measurement matrix $\mathbf{M}$ (top row in Figure 2)

Notation:

$\mathbf{m}_{ij} = [u_{i,j}, v_{i,j}, 1]^T$, $\omega_{ij} \in [0, 1]$.
$\mathbf{M}_{sub} \in \mathbb{R}^{3p\times q}$ and $\Omega_{sub} \in \mathbb{R}^{2p\times q}$ are sub-matrices of $\mathbf{M}$ and Ω, respectively. η is an expected ratio of known entries. ρ is an expected confidence threshold of Ω.

1: Initialize the mask matrix Ω by using Eq. (6).
2: Select some $\mathbf{M}_{sub}$'s whose Ω_{sub}'s satisfy the following conditions:
$\sum_{i=1}^{p}\{\omega_{sub}\}_{ij} > p\times\eta, \forall j$, $\sum_{j=1}^{q}\{\omega_{sub}\}_{ij} > q\times\eta, \forall i$, i.e., the ratios of known entries in every row and column are above the threshold η.
3: Recover all the $\mathbf{M}_{sub}$'s in parallel by solving Eq. (5). Store each $\widehat{\mathbf{M}}_{sub}$ as a candidate value of $\mathbf{m}_{ij}$ (Figure 3).
4: Update $\omega_{ij} = \exp\left(-\frac{\sum_{k=1}^{n}\|\mathbf{m}_k-\overline{\mathbf{m}}\|_2}{n}\right)$, where n is the total number of all candidate values, $\mathbf{m}_k$ is the k^{th} candidate value of $\mathbf{m}_{ij}$, and $\overline{\mathbf{m}}$ is the median of $\{\mathbf{m}_k\}$.
5: If $\omega_{ij} > \rho$, then update $\mathbf{M}$: $\mathbf{m}_{ij} = \overline{\mathbf{m}}$.
6: If $\mathbf{M}$ is not fully filled-in, then go to step 2.
7: **Output:**
Fully filled-in measurement matrix $\mathbf{M}$ (bottom row in Figure 2).

After recovering the rescaled measurement matrix $\mathbf{W}$, a projective reconstruction is available to apply by directly factorizing $\mathbf{W}$ into $\widehat{\mathbf{P}}\widehat{\mathbf{X}}$, where $\widehat{\mathbf{P}} = [\widehat{\mathbf{P}}_1; \widehat{\mathbf{P}}_2; \cdots; \widehat{\mathbf{P}}_m] \in \mathbb{R}^{3m\times 4}$ is the camera motion matrix and $\widehat{\mathbf{X}} \in \mathbb{R}^{4\times n}$ is the scene structure matrix.

C. Upgrade to Metric reconstruction

Clearly, the projective factorization is not unique: $\mathbf{W} = (\widehat{\mathbf{P}}\mathbf{H})(\mathbf{H}^{-1}\widehat{\mathbf{X}})$, where $\mathbf{H} \in \mathbb{R}^{4\times 4}$ is any non-singular matrix. In real applications, a metric reconstruction is more desirable as it differs from the true reconstruction only by a similarity transformation. So we upgrade the projective reconstruction to the metric reconstruction by reducing the ambiguity in $\mathbf{H}$. To this end, we assume that the intrinsic parameters of all the cameras are the same.

Based on the projective imaging model, the relation between the ground truth $\mathbf{P}$ and the intermediate $\widehat{\mathbf{P}}$ can be expressed as [5]:

$$\widehat{\mathbf{P}}_i\mathbf{H} = \alpha_i\mathbf{P}_i = \alpha_i\mathbf{K}[\mathbf{R}_i\ \mathbf{t}_i], \quad i = 1, \cdots, m,$$

where $\mathbf{K}$ is an upper-triangular matrix composed of camera intrinsic parameters, $\mathbf{R}_i$ and $\mathbf{t}_i$ are the rotation and translation of the i-th camera, respectively.

Rewriting $\mathbf{H}$ as $[\mathbf{H}_3, \mathbf{h}]$ where $\mathbf{H}_3 \in \mathbb{R}^{4\times 3}$ consists of the first three columns of $\mathbf{H}$, we have

$$\widehat{\mathbf{P}}_i\mathbf{H}_3 = \alpha_i\mathbf{K}\mathbf{R}_i, \quad i = 1, \cdots, m,$$

and hence [18]:

$$\widehat{\mathbf{P}}_i\mathbf{H}_3\mathbf{H}_3^T\widehat{\mathbf{P}}_i^T = \alpha_i^2\mathbf{K}\mathbf{K}^T, \quad i = 1, \cdots, m.$$

Let $\mathbf{Q} \equiv \mathbf{H}_3\mathbf{H}_3^T$, which is positive semi-definite of rank 3, and $\mathbf{V} \equiv \mathbf{K}\mathbf{K}^T$, which is positive definite and called the dual image of the absolute conic (DIAC) [1]. Using notation $\mathbf{V}_i \equiv \alpha_i^2\mathbf{K}\mathbf{K}^T$, we obtain

$$\widehat{\mathbf{P}}_i\mathbf{Q}\widehat{\mathbf{P}}_i^T = \mathbf{V}_i, \quad i = 1, \cdots, m. \tag{7}$$

Eq. (7) can be rewritten as:

$$(\widehat{\mathbf{P}}_i \otimes \widehat{\mathbf{P}}_i)\mathrm{vec}(\mathbf{Q}) = \mathrm{vec}(\mathbf{V}_i), \quad i = 1, \cdots, m,$$

where $\mathrm{vec}(\cdot)$ is the vectorization operator that stacks all the columns of a matrix into a long column vector. Notice that the rank of $\overline{\mathbf{V}} \equiv [\mathrm{vec}(\mathbf{V}_1), \cdots, \mathrm{vec}(\mathbf{V}_m)]$ is one, as each column is a scaled $\mathrm{vec}(\mathbf{V})$.

Combining all constraints together, we have the following optimization problem for $\mathbf{Q}$ and $\mathbf{V}_i$:

$$\begin{aligned}\min_{\mathbf{Q},\{\mathbf{V}_i\}} \quad & \sum_{i=1}^{m}\left\|(\widehat{\mathbf{P}}_i \otimes \widehat{\mathbf{P}}_i)\mathrm{vec}(\mathbf{Q}) - \mathrm{vec}(\mathbf{V}_i)\right\|_F^2 \\ & + \|\mathbf{Q}\|_{*,r_2} + \|\overline{\mathbf{V}}\|_{*,r_3}, \\ \text{s.t.} \quad & \mathbf{Q} \succeq \mathbf{0}, \mathbf{V}_1 \succeq \mathbf{I}, \mathbf{V}_i \succeq \mathbf{0}, i = 2, \cdots, m,\end{aligned} \tag{8}$$

where $r_2 = 3$, $r_3 = 1$, and the positive semi-definite constraint $\mathbf{V}_1 \succeq \mathbf{I}$ is imposed to avoid trivial solutions $\mathbf{Q} = \mathbf{0}$ and $\overline{\mathbf{V}} = \mathbf{0}$. Eq. (8) can be solved by applying ADMM [17] and semidefinite programming (SDP) [19]. We will provide the details on our website or arXiv later.

When $\mathbf{Q}$ is obtained, $\mathbf{H}_3$ can be directly computed by the eigenvalue decomposition of $\mathbf{Q}$ with a rotation transformation ambiguity. As $\mathbf{h}$ is the homogeneous coordinate of the origin of the world coordinate, an arbitrary $\mathbf{h}$ can be selected as long as $\mathbf{H} = [\mathbf{H}_3, \mathbf{h}]$ is non-singular. After recovering $\mathbf{H}$, the metric reconstruction is achieved by

$$\mathbf{P} = \widehat{\mathbf{P}}\mathbf{H} \text{ and } \mathbf{X} = \mathbf{H}^{-1}\widehat{\mathbf{X}}.$$

The intrinsic parameter $\mathbf{K}$ can be obtained from the Cholesky decomposition of $\mathbf{V}_1$ with a scale transformation ambiguity.

III. Optimization by ADMM

In this paper we adopt the alternating direction method of multiplier (ADMM) [17] to solve our truncated nuclear norm minimization problems. ADMM is a representative first order method due to its simplicity and has found to be very effective for many nuclear norm minimization problems [13], [17].

To make each sub-problem of ADMM have a closed-form solution, we introduce an auxiliary matrix $\mathbf{Z}$ and rewrite problem Eq. (5) as:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{E},\mathbf{Z}} \quad & \|\mathbf{W}\|_{*,r_1} + \hat{\tau}\|\mathbf{E}\|_{\Omega\odot 1}, \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{Z}) + \mathbf{E} = \mathbf{0},\ \mathbf{W} - \mathbf{Z} = \mathbf{0},\ \mathbf{d}^T\mathbf{Z}\mathbf{1} = mn. \end{aligned} \tag{9}$$

The partial augmented Lagrangian function of Eq. (9) is:

$$\begin{aligned} \mathcal{L}_1(\mathbf{W},\mathbf{E},\mathbf{Z},\mathbf{G},\mathbf{L}) = & \|\mathbf{W}\|_{*,r_1} + \hat{\tau}\|\mathbf{E}\|_{\Omega\odot 1} \\ & + \langle \mathbf{G}, \mathcal{A}(\mathbf{Z}) + \mathbf{E}\rangle + \tfrac{\alpha}{2}\|\mathcal{A}(\mathbf{Z}) + \mathbf{E}\|_F^2 \\ & + \langle \mathbf{L}, \mathbf{W} - \mathbf{Z}\rangle + \tfrac{\alpha}{2}\|\mathbf{W} - \mathbf{Z}\|_F^2, \end{aligned} \tag{10}$$

where $\mathbf{G}$ and $\mathbf{L}$ are the Lagrange multiplier matrices, $\langle \mathbf{A}, \mathbf{B}\rangle \equiv \text{trace}(\mathbf{A}^T\mathbf{B})$ denotes the matrix inner product, and $\alpha > 0$ is a penalty parameter.

The iterations of ADMM go as follows:

$$\begin{aligned} \mathbf{W}_{k+1} &= \arg\min_{\mathbf{W}} \mathcal{L}_1(\mathbf{W},\mathbf{E}_k,\mathbf{Z}_k,\mathbf{G}_k,\mathbf{L}_k) \\ &= \arg\min_{\mathbf{W}} \|\mathbf{W}\|_{*,r_1} + \tfrac{\alpha}{2}\|\mathbf{W} - \mathbf{Z}_k + \tfrac{\mathbf{L}_k}{\alpha}\|_F^2, \end{aligned} \tag{11}$$

$$\begin{aligned} \mathbf{E}_{k+1} &= \arg\min_{\mathbf{E}} \mathcal{L}_1(\mathbf{W}_{k+1},\mathbf{E},\mathbf{Z}_k,\mathbf{G}_k,\mathbf{L}_k) \\ &= \arg\min_{\mathbf{E}} \hat{\tau}\|\mathbf{E}\|_{\Omega\odot 1} + \tfrac{\alpha}{2}\|\mathbf{E} + \mathcal{A}(\mathbf{Z}_k) + \tfrac{\mathbf{G}_k}{\alpha}\|_F^2, \end{aligned} \tag{12}$$

$$\begin{aligned} \mathbf{Z}_{k+1} &= \arg\min_{\mathbf{d}^T\mathbf{Z}\mathbf{1}=mn} \mathcal{L}_1(\mathbf{W}_{k+1},\mathbf{E}_{k+1},\mathbf{Z},\mathbf{G}_k,\mathbf{L}_k) \\ &= \arg\min_{\mathbf{d}^T\mathbf{Z}\mathbf{1}=mn} \tfrac{\alpha}{2}\|\mathcal{A}(\mathbf{Z}) + \mathbf{E}_{k+1} + \tfrac{\mathbf{G}_k}{\alpha}\|_F^2 \\ &\quad + \tfrac{\alpha}{2}\|\mathbf{W}_{k+1} - \mathbf{Z} + \tfrac{\mathbf{L}_k}{\alpha}\|_F^2, \end{aligned} \tag{13}$$

$$\mathbf{G}_{k+1} = \mathbf{G}_k + \alpha(\mathcal{A}(\mathbf{Z}_{k+1}) + \mathbf{E}_{k+1}), \tag{14}$$

$$\mathbf{L}_{k+1} = \mathbf{L}_k + \alpha(\mathbf{W}_{k+1} - \mathbf{Z}_{k+1}). \tag{15}$$

According to Lu et al.'s proof [20], the solution to Eq. (11) is given by:

$$\mathbf{W}_{k+1} = \mathcal{D}_{r_1,\alpha^{-1}}\left(\mathbf{Z}_k - \frac{\mathbf{L}_k}{\alpha}\right), \tag{16}$$

where $\mathcal{D}_{r,\varepsilon}(\mathbf{Y})$ ($\varepsilon > 0$) is the truncated singular value shrinkage operator defined as follows:

$$\mathcal{D}_{r,\varepsilon}(\mathbf{Y}) \equiv \mathbf{U}(\Sigma - \varepsilon\text{diag}(\mathbf{c}))_+\mathbf{V}^T,$$

in which $\mathbf{U}\Sigma\mathbf{V}^T$ is the SVD of $\mathbf{Y}$, $(x)_+ = \max\{x, 0\}$, and $\mathbf{c}$ is a vector where $c_1 = \cdots = c_r = 0$ and $c_{r+1} = \cdots = c_{\min(p,q)} = 1$.

The solution to Eq. (12) is

$$\mathbf{E}_{k+1} = \mathcal{S}_{\Omega,\alpha^{-1}}\left(-\mathcal{A}(\mathbf{Z}_k) - \frac{\mathbf{G}_k}{\alpha}\right), \tag{17}$$

where $\mathcal{S}_{\Omega,\varepsilon}(\mathbf{Y})$ is the weighted shrinkage operator:

$$\mathcal{S}_{\Omega,\varepsilon}(\mathbf{Y}) \equiv (|\mathbf{Y}| - \varepsilon\Omega)_+ \odot \text{sgn}(\mathbf{Y}),$$

in which $|\mathbf{Y}| = [|y_{ij}|]$. The proof of Eq. (17) is in [21].

Here we propose an extremely efficient algorithm to solve Eq. (13), which is hard to solve directly. In detail, denote the matrix representation of the operator $\mathcal{A}$ as $\mathbf{A}$, where $\mathbf{A} = \text{diag}(\cdots, \mathbf{A}_{ij}, \cdots)$, and $\mathbf{A}_{ij} = \begin{bmatrix} 1 & 0 & -u_{ij} \\ 0 & 1 & -v_{ij} \end{bmatrix}$, then the optimality condition for Eq. (13) is

$$\begin{aligned} & \begin{bmatrix} \alpha(\mathbf{I} + \mathbf{A}^T\mathbf{A}) & \widetilde{\mathbf{D}} \\ \widetilde{\mathbf{D}}^T & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{Z}} \\ \beta \end{bmatrix} \\ = & \begin{bmatrix} \alpha\widetilde{\mathbf{W}}_{k+1} + \widetilde{\mathbf{L}}_k - \mathbf{A}^T(\alpha\widetilde{\mathbf{E}}_{k+1} + \widetilde{\mathbf{G}}_k) \\ mn \end{bmatrix}, \end{aligned} \tag{18}$$

where β is an introduced Lagrange multiplier, $\widetilde{[\cdot]}$ means the vectorized one of the matrix, and $\mathbf{D} = \mathbf{1}^T \otimes \mathbf{d}$. A direct way to solve Eq. (18) will involve extremely expensive computation, because the size of the coefficient matrix is $(9mn + 1) \times (9mn + 1)$. We first compute the eigenvalue decomposition of $\mathbf{A}_{ij}^T\mathbf{A}_{ij}$ as $\mathbf{V}_{ij}\mathbf{S}_{ij}\mathbf{V}_{ij}^T$, which has a closed-form solution. Denote $\mathbf{S} = \text{diag}(\cdots, \mathbf{S}_{ij}, \cdots)$ and $\mathbf{V} = \text{diag}(\cdots, \mathbf{V}_{ij}, \cdots)$, then

$$\begin{aligned} & \begin{bmatrix} \mathbf{V} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha(\mathbf{I} + \mathbf{A}^T\mathbf{A}) & \widetilde{\mathbf{D}} \\ \widetilde{\mathbf{D}}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & 0 \\ 0 & 1 \end{bmatrix} \\ = & \begin{bmatrix} \alpha(\mathbf{I} + \mathbf{S}) & \mathbf{V}\widetilde{\mathbf{D}} \\ (\mathbf{V}\widetilde{\mathbf{D}}^T) & 0 \end{bmatrix} \end{aligned}$$

is an *arrowhead* matrix, whose non-zero entries are only on the last row, the last column, and the diagonal. Let $\mathbf{x} = \begin{bmatrix} \mathbf{V}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{Z}} \\ \beta \end{bmatrix}$, and $\mathbf{y} = \begin{bmatrix} \mathbf{V}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha\widetilde{\mathbf{W}}_{k+1} + \widetilde{\mathbf{L}}_k - \mathbf{A}^T(\alpha\widetilde{\mathbf{E}}_{k+1} + \widetilde{\mathbf{G}}_k \\ mn \end{bmatrix}$, then the Eq. (18) can be transformed into a much simpler linear system:

$$\begin{bmatrix} \text{diag}(\mathbf{a}) & \mathbf{b} \\ \mathbf{b}^T & 0 \end{bmatrix} \mathbf{x} = \mathbf{y}, \tag{19}$$

where $\text{diag}(\mathbf{a}) \in \mathbb{R}^{9mn}$ is the diagonal of $\alpha(\mathbf{I} + \mathbf{S})$ and $\mathbf{b} = \mathbf{V}\widetilde{\mathbf{D}} \in \mathbb{R}^{9mn}$. Since $\mathbf{V}$ and $\mathbf{A}^T$ are block diagonal, their multiplication with a vector can be efficiently calculated. Finally, the solution to Eq. (19) is simply given as:

$$\begin{aligned} x_{9mn+1} &= \left[\sum_{j=1}^{9mn}(b_j y_j/a_j) - y_{9mn+1}\right] \Big/ \sum_{j=1}^{9mn+1}(b_i^2/a_i), \\ x_i &= (y_i - b_i x_{9mn+1})/a_i, \quad 1 \le i \le 9mn. \end{aligned}$$

After $\mathbf{x}$ is computed, we can compute $\mathbf{Z}$ by solving

$$\begin{bmatrix} \widetilde{\mathbf{Z}} \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{V} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}.$$

IV. Experiments

In this section, we conduct extensive experiments on both synthetic and real-world datasets to evaluate the performance of our factorization methods for projective and metric reconstructions.

For projective reconstruction, we cannot compare with [7] because their method requires a smooth and densely sampled trajectory of the camera. Two measures are employed to evaluate the performance:

(1) *Reprojection errors* in the image space [6]:

$$\varepsilon_1 \equiv \|\mathbf{m} - \widehat{\mathbf{m}}\|_2,$$

where $\mathbf{m} \in \mathbb{R}^2$ is an imaged point and $\widehat{\mathbf{m}}$ is its reprojected point that is computed by $\widehat{\mathbf{m}} = [w_{3i-2,j}/w_{3i,j}, w_{3i-1,j}/w_{3i,j}]^T$ (for the j-th point in the i-th image).

(2) *Relative 3D reconstruction errors* in the 3D space:

$$\varepsilon_2 \equiv \|\mathbf{x} - \widehat{\mathbf{x}}\|_2/\|\mathbf{x}\|_2,$$

where $\mathbf{x} \in \mathbb{R}^3$ is the ground truth 3D point and $\widehat{\mathbf{x}}$ is the estimation.

For metric reconstruction, we compare our method with the classic SfM method named "Bundler", which was proposed by Snavely et al. [22]. We use the following measure for performance evaluation:

Overall relative 3D reconstruction error :

$$\varepsilon_3 = \|\widetilde{\mathbf{X}} - \mathbf{X}^t\|_F/\|\mathbf{X}^c\|_F,$$

where $\mathbf{X}^c$ is the matrix of centered 3D points by deducting their mean from the ground truth $\mathbf{X}^t$ and $\widetilde{\mathbf{X}}$ is the reconstruction output corrected by resolving scale, rotation, and reflection ambiguities using $\mathbf{X}^t$ as a reference.

A. Experiments with synthetic data

We test the robustness of our algorithm under different settings. For all the synthetic experiments, we set the camera intrinsic matrix as the identity matrix.

1) Accuracy in projective reconstruction: We first test our algorithm on synthetic data with both missing data and outliers. We add different ratios of outliers, ranging from 0% to 10% to the synthetic data, together with 20% uniformly random missing data. Empirically, the ratio of outliers caused by mismatching is below 5% in a real stereo system, so our settings of outliers are sufficient. The best τ is selected for different settings. As shown in Figure 4, our algorithm works well in a wide variety of scale settings with both missing data and outliers.

2) Accuracy in metric reconstruction: We use 3D Max to simulate a 3D point cloud of a house observed from 10 or 20 cameras (Figures 5(a)) and with dense noise of ± 0.5 pixels. The missing data are naturally produced by occlusion by a teapot and an ellipsoid. Different ratios of outliers (up to 10%) are further imposed. The metric reconstruction results of Snavely et al.'s classic SfM method "Bundler" [22] and our method are shown in Figures 5(b) and 5(c), whose overall relative 3D reconstruction errors are $\varepsilon_3 = 5.93\%$ and 0.47%, respectively. Our method is clearly much more accurate than "Bundler" [22]. Moreover, "Bundler" cannot recover all the points because it does not handle missing values well.

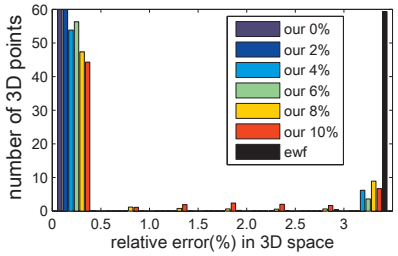
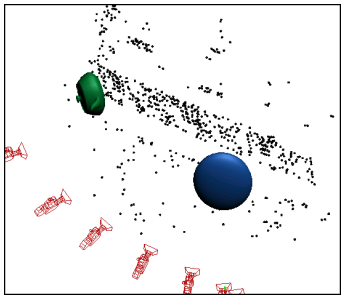
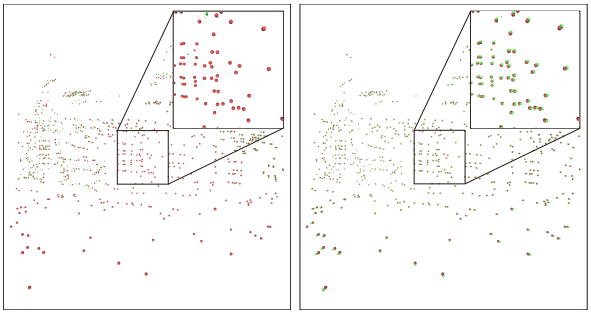


Fig. 4. Histograms of relative 3D reconstruction errors on synthetic data with 20 cameras × 60 points. The percentage of missing data is set to 20% with different percentages of outliers. In our algorithm, τ's are tuned to achieve the best performance ($\tau = 0.35$).



(a)



(b) (c)

Fig. 5. Results on data simulated by 3D Max. (a) A simulated scene consisting of a point cloud of a house and two obstacles (teapot and ellipsoid), captured by several cameras. (b) Overlap of the ground truth (red sphere) and the 3D points (green square) reconstructed by using Bundler [22]. *It cannot recover all of the points.* (c) Overlap of the ground truth (red sphere) and the estimated 3D points (green square) by our metric reconstruction.

B. Experiments with real data

1) Accuracy in projective reconstruction: **Comparison with state-of-the-art methods.** First, we compare the average reprojection errors on two real datasets: Corridor and Chair [6]. Table I shows that the reprojection errors of our algorithm are much lower than that of the state-of-the-art methods on both datasets. Due to memory limitation, L1-Wiberg [8] fails on the Corridor dataset.

Robustness to corruptions. We also verify our method on the first seven frames of Oxford Dinosaur dataset [25].

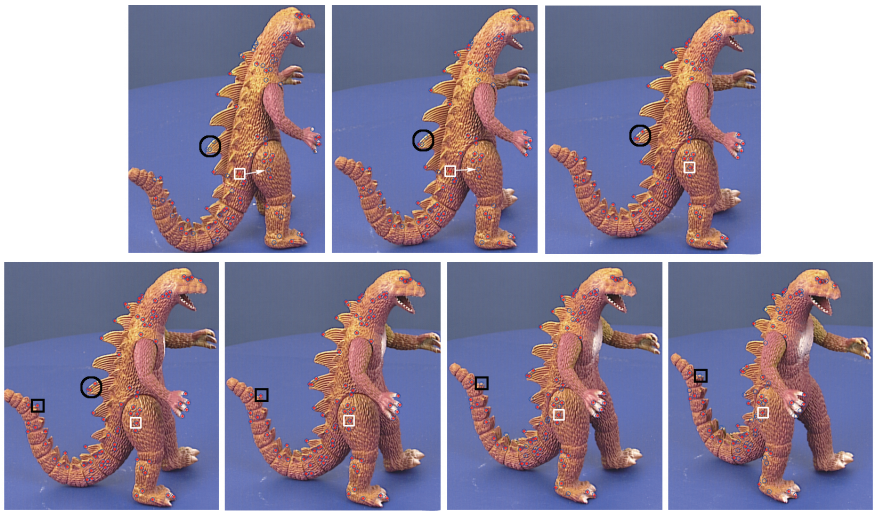


Fig. 6. Results on the Oxford dinosaur data by using our algorithm. Most of the input points (red star) coincide with the estimated reprojected points (blue circles). Black circles and black squares show examples of missing data in the original input points, which are recovered by our algorithm. White squares indicate one of the obvious outliers (No.578 point in the input data) found by our algorithm, which has never been reported before.

TABLE I
COMPARISON OF AVERAGE REPROJECTION ERRORS ON REAL IMAGES WITH COMPLETE MEASUREMENTS (SOME ARE QUOTED FROM [6]).

Method/Dataset (cameras×points)	Corridor (11×104)	Chair (6×9)
SIESTA [23]	0.4328	1.3584
CIESTA [23]	0.4296	1.3723
Col-space [24]	0.4501	1.4759
EWF [6]	0.4327	1.3568
EWF [6]+Bundler [22]	0.3763	1.2545
L1-Wiberg [8]	-	0.9368
Our algorithm	**0.2823**	**0.9154**

Figure 6 shows the ground truth and the recovered reprojected points which are visible in at least 4 frames. We can see that our algorithm can handle missing data and outliers simultaneously. Notice that our model does *not* need to explicitly distinguish missing data from outliers.

2) Accuracy in metric reconstruction: Since the Oxford Dinosaur dataset has ground truth, we seek to recover metric reconstruction of the whole dinosaur and then compare with the ground truth. Unfortunately, due to the occlusions, the whole dataset of 36 frames yields a strong band-diagonal pattern in the measurement matrix with almost all off-band-diagonal entries missing, which is regarded as a very challenging problem for 3D recovery [7]. So the first step of this experiment is to recover the whole measurement matrix. To this end, we first select points which appear in more than four frames. So there are more than 900 points in the whole measurement matrix. Next, we fill in the original measurement matrix by using Algorithm 1. Figure 7 shows an example of updating missing points by assigning the median values to it.

With the fully filled-in measurement matrix, we apply our metric reconstruction method to it. Figure 8(b) shows the 3D points collected by the measurement matrix (more than 900 points). After aligning (i.e., by removing the scale and rotation/reflection ambiguities) our metric reconstruction

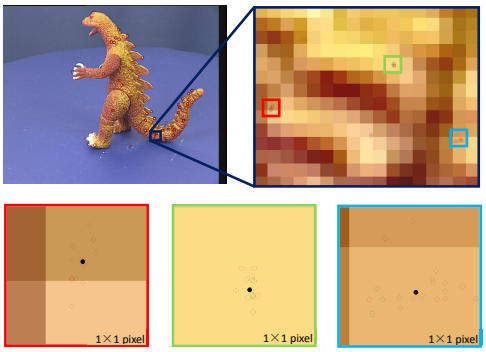
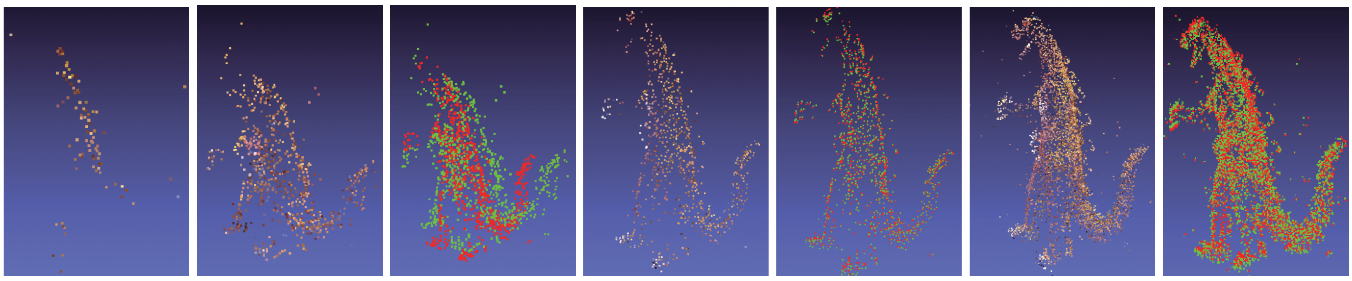


Fig. 7. A example of updating the positions of missing points. Top left: a frame with all recovered missing points in it. Top right: all the candidate values of one missing point are in the same rectangle. Bottom row: different colors of points stand for different candidate values. The median value of all the candidate values is shown as the black point.

with the ground truth, which is obtained by triangulation using the camera matrices provided by the dataset [25], we can see that they match fairly well (Figure 8(e)). Actually, the overall relative 3D reconstruction error is only 6.18%. We also apply the estimated camera matrix $\mathbf{P}$ to all the points recorded in the dataset [25] (about 5000 points) and obtain the triangulation reconstruction, as shown in Figure 8(f). Again, our triangulation reconstruction and the ground truth model match well (Figure 8(g)). In contrast, "Bundler" [22] can only reconstruct part of the points (Figures 8(a) and 8(c)) and the error is much larger (40.80%, Figure 8(d)). This shows the effectiveness of our metric reconstruction method. In particular, we want to re-emphasize that the original measurement matrix is very sparse and we have to fill in it first by Algorithm 1. This strongly proves the high robustness of our method.

V. CONCLUSIONS

In this paper we argue that the commonly used nuclear norm is ineffective in producing fixed rank solutions for SfM problems. We then apply recently proposed truncated



(a) (b) (c) (d) (e) (f) (g)

Fig. 8. (a) Reconstruction by Bundler [22] from a subset of the original measurement matrix (more than 900 points). (b) Reconstruction by Bundler from our fully filled-in measurement matrix. (c) Registration between the reconstruction by Bundler (green points) and the ground truth triangulation reconstruction (red points) computed by using the camera matrices provided in the dataset [25]. The overall relative 3D error is about **40.80%**. (d) Metric reconstruction by our method from the same subset as (a). (e) Registration between our metric reconstruction (green points) and the ground truth triangulation reconstruction (red points). The overall relative 3D error is about **6.18%**. (f) Triangulation reconstruction of all points in the data set (about 5000 points) by utilizing our estimated camera matrix $\mathbf{P}$. (g) Registration between our triangulation reconstruction (green points) and the ground truth (red points).

nuclear norm to better enforce the fixed rank constraints while preserving the elegance of the factorization framework. Based on truncated nuclear norm, we propose the projective reconstruction and metric upgrade methods to pursuit more accurate 3D reconstructions than the state-of-the-arts. Moreover, our method deals with outliers and missing data in a unified way by using weighted ℓ_1-norm. We also provide an extremely efficient algorithm to tackle one of the optimization sub-problems and get better reconstruction result. Extensive experiments on synthetic and real-world data have demonstrated the efficacy and robustness of our method.

ACKNOWLEDGMENT

Zhouchen Lin was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2015CB352502, in part by the National Natural Science Foundation (NSF) of China under Grant Nos. 61625301 and 61231002, and in part by the Qualcomm. Tong Lin was supported by the NSF of China under Grant No. 61375051. Hongbin Zha was supported by Beijing Municipal Natural Science Foundation under Grant No. 4152006.

REFERENCES

[1] A. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (2nd ed.)*. Cambridge University Press, 2004.

[2] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach (2nd ed.)*. Prentice Hall, 2011.

[3] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137–154, 1992.

[4] B. Triggs, "Factorization methods for projective structure and motion," in *Computer Vision and Pattern Recognition*, 1996.

[5] T. Ueshiba and F. Tomita, "A factorization method for projective and Euclidean reconstruction from multiple perspective views via iterative depth estimation," in *European Conference on Computer Vision*, 1998.

[6] Y. Dai, H. Li, and M. He, "Projective multiview structure and motion from element-wise factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2238–2251, 2013.

[7] R. Angst, C. Zach, and M. Pollefeys, "The generalized trace-norm and its application to structure-from-motion problems," in *International Conference on Computer Vision*, 2011.

[8] A. Eriksson and A. van den Hengel, "Efficient computation of robust weighted low-rank matrix approximations using the l_1 norm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1681–1690, 2012.

[9] S. Maybank and O. Faugeras, "A theory of self-calibration of a moving camera," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 123–151, 1992.

[10] P. R. S. Mendonca and R. Cipolla, "A simple technique for self-calibration," in *Computer Vision and Pattern Recognition*, 1999.

[11] M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown instrinsic camera parameters," *International Journal of Computer Vision*, vol. 32, pp. 7–25, 1999.

[12] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2117–2130, 2013.

[13] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, p. 11, 2011.

[14] Z. Zhang, S. Yan, and M. Zhao, "Similarity preserving low-rank representation for enhanced data representation and effective subspace learning," *Neural Networks*, vol. 53, pp. 81–94, 2014.

[15] Z. Zhang, S. Yan, M. Zhao, and F. Li, "Bilinear low-rank coding framework and extension for robust image recovery and feature representation," *Knowledge-Based Systems*, vol. 86, no. C, pp. 143–157, 2015.

[16] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Joint low-rank and sparse principal feature coding for enhanced robust representation and visual classification," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2429–2443, 2016.

[17] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, 2011.

[18] G. Wang and Q. J. Wu, "Quasi-perspective projection model: theory and application to structure and motion factorization from uncalibrated image sequences," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 213–234, 2010.

[19] S. J. Benson and Y. Ye, "DSDP5: Software for semidefinite programming," Tech. Rep., 2005.

[20] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, "Generalized singular value thresholding," in *AAAI Conference on Artificial Intelligence*, 2015.

[21] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.

[22] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[23] J. Oliensis and R. Hartley, "Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2217–2233, 2007.

[24] S. Mahamud and M. Hebert, "Iterative projective reconstruction from multiple views," in *Computer Vision and Pattern Recognition*, 2000.

[25] A. M. Buchanan and A. W. Fitzgibbon, "Damped newton algorithms for matrix factorization with missing data," in *Computer Vision and Pattern Recognition*, 2005.