Element-wise Factorization for N-view Projective Reconstruction

Yuchao Dai[1,2], Hongdong Li[3,2], and Mingyi He[1]

[1] School of Electronics and Information, Northwestern Polytechnical University
Shaanxi Key Laboratory of Information Acquisition and Processing, Xi'an China
[2] Australian National University, Australia
[3] Canberra Research Lab, NICTA, Australia

Abstract. Sturm-Triggs iteration is a standard method for solving the *projective factorization* problem. Like other iterative algorithms, this method suffers from some common drawbacks such as requiring a good initialization, the iteration may not converge or only converge to a local minimum, etc. None of the published works can offer any sort of global optimality guarantee to the problem. In this paper, an optimal solution to projective factorization for structure and motion is presented, based on the same principle of low-rank factorization. Instead of formulating the problem as *matrix factorization*, we recast it as *element-wise factorization*, leading to a convenient and efficient semi-definite program formulation. Our method is thus *global*, where no initial point is needed, and a globally-optimal solution can be found (up to some relaxation gap). Unlike traditional projective factorization, our method can handle real-world difficult cases like missing data or outliers easily, and all in a unified manner. Extensive experiments on both synthetic and real image data show comparable or superior results compared with existing methods.

1 Introduction

Tomasi-Kanade factorization [1] is probably one of the most remarkable works in multi-view structure-from-motion (SFM) research. This algorithm is not only of significant theoretical importance, but also of striking elegance and computational simplicity. Given a multi-view measurement matrix $\mathtt{M}$, it simultaneously solves for the (stacked) camera projection matrix $\mathtt{P}$ and the 3D structure $\mathtt{X}$, via a simple *matrix factorization* $\mathtt{M} = \mathtt{PX}$ through a single Singular Value Decomposition (SVD). Its elegance also comes from the fact that it treats all points and all camera frames *uniformly*, no any "privileged" or "preferred" frames and points.

Tomasi-Kanade's factorization algorithm was developed for *affine* camera cases. We revisit the *projective* generalization of the factorization method in this paper. In particular, we are motivated by the most popular algorithm-of-choice for *projective factorization*–the iterative Sturm-Triggs method[4] [2][3]. Projective

[4] We exclude the non-iterative version of Sturm-Triggs method reported in [2], as it crucially relies on accurate depth estimation from epipolar geometry.

imaging process can be compactly written as $\mathtt{\Lambda} \odot \mathtt{M} = \mathtt{PX}$ where $\mathtt{\Lambda}$ matrix is a properly stacked but unknown *projective depth matrix*, $\odot$ denotes the element-wise (Hadamard) product. Since both the depths $\mathtt{\Lambda}$ and the right-hand factorization $\mathtt{P}$ and $\mathtt{X}$ are unknowns, a natural approach to solve this is through iterative alternation till converge. Many other projective generalizations, such as [4] [5] [6] share a similar computational pattern in terms of iteration.

Like any other iterative algorithm, the Sturm-Triggs iteration and its extensions have some common drawbacks. For example, iterative algorithms all require a good initial point to start with; the iteration procedure may not converge; even if it does converge (theoretically or empirically), it may only converge to a local minimum; global optimality can hardly be guaranteed.

Unfortunately, for the particular method of Sturm-Triggs iteration and the alikes, all the aforementioned drawbacks did have been observed in all occasions. Indeed, [7] pointed out that the iterative Sturm-Triggs method with row- and column- normalization is not guaranteed to converge in theory. To salvage this, they proposed a column-wise only normalization and derived a provably-convergent iterative algorithm (called *column-space* method) [7]. Oliensis and Hartley also observed situations where the iterations fell into a limiting cycle and never converged [8]. Hartley and Zisserman [9] concluded that the popular choice of initialization–assuming all depths to be one–works only when the ratios of true depths of the different 3D point $\mathbf{X}_j$ remain approximately constant during a sequence. As a result, to make Sturm-Triggs iteration work, the true solution has to be rather close to the affine case.

Even worse, a recent complete theoretical analysis delivers even more negative message [8], which shows that (1) the simplest Sturm-Triggs iteration without normalization (called SIESTA w.o. balancing) will always converge to the trivial solution; (2) paper [10]'s provably-convergent iteration method will generally converge to a useless solution; (3) applying both row-wise and column-wise normalization may possibly run into unstable state during iteration. The authors also provided a remedy, i.e. a regularization-based iterative algorithm (called CIESTA) that can converge to a stable solution, albeit the solution is biased (towards all depths being close to one).

Having mentioned the above negative points, we however argue that Sturm-Triggs algorithm (and its variants) are useful in practice. After a few iterations they often return a much improved and useful result. In fact, many of the issues discussed in [8] are theoretically driven. However when the actual camera-point configuration is far away from affine configuration, Sturm-Triggs algorithm tends to produce a bad solution. It would be nicer if a projective-factorization algorithm can be made free from these theoretical drawbacks and can at the same time be useful in practice.

In this paper, we propose a closed-form solution to projective factorization, which is based on the similar idea of low-rank factorization and stays away from all the above mentioned theoretical traps. Our algorithm is global; no initial guess is needed. Given a complete measurement matrix the result will be globally optimal (at most up to some relaxation gap). Additionally, it deals with missing data and outliers all in unified framework. The outlier-extension of our method has an intimate connection with the recent proposed Compressive Sensing theory and algorithms. Nevertheless, the main theory and algorithms of our method stand independently, and do not depend on compressive sensing theory.

2 Element-Wise Factorization

2.1 Preliminaries

Consider n stationary 3D points $\mathbf{X}_j = [x_j, y_j, z_j, 1]^T, j = 1, \cdots, n$ observed by all m projective cameras $\mathtt{P}_i, i = 1, \cdots, m$. Under projective camera model, the j-th 3D point $\mathbf{X}_j$ is projected onto image point $\mathbf{m}_{ij} = [u_{ij}, v_{ij}, 1]^T$ by $\mathbf{m}_{ij} = \frac{1}{\lambda_{ij}}\mathtt{P}_i\mathbf{X}_j$, where λ_{ij} is a scale factor, commonly called "projective depth" [9]. It is easy to see that $\lambda_{ij} = 1$ when the camera reduces to an affine camera.

Collecting all the image measurements over all frames, we form a *measurement matrix* $\mathtt{M} = [\mathbf{m}_{ij}]$ of size $(3m \times n)$. Now the above relationship is compactly written as

$$\mathtt{M} = \left[(\frac{1}{\lambda_{ij}})\right]_\sharp \odot (\mathtt{PX}), \tag{1}$$

where $\mathtt{P} \in \mathbb{R}^{3m\times 4}$ and $\mathtt{X} \in \mathbb{R}^{4\times n}$ are properly stacked projection matrix and structure matrix. Note that each row of the inverse depth matrix is repeated 3 times. We use a subscript of "$\sharp$" to denote such a triple copy.

Define $\mathtt{W} = \mathtt{PX}$ as the **rescaled (re-weighted) measurement matrix**, we can equivalently re-write the above equation as:

$$\mathtt{W} = \Lambda \odot \mathtt{M} = \mathtt{PX}, \tag{2}$$

where $\Lambda = [(\lambda_{ij})]_\sharp \in \mathbb{R}^{3m\times n}$, i.e. a triple copy. As seen from the equation, matrix $\mathtt{W}$ must have a rank at most 4.

The problem of projective factorization seeks to simultaneously solve for the unknown depths Λ, the unknown cameras $\mathtt{P}$ and the unknown structure $\mathtt{X}$. Compared with affine factorization, this is a much harder problem, mainly because depths are not known *a priori*. Of course, one could compute these projective depths beforehand, by other means, e.g, via fundamental matrices or trifocal tensors, via a common reference plane, etc. However, such approaches diminish the elegance of the factorization algorithm, as they no longer treat points and frames uniformly.

The Sturm-Triggs type iterative algorithms solve the problem through *alternation*: (1) fix Λ, solve for $\mathtt{P}$ and $\mathtt{X}$ via SVD factorization; (2) fix $\mathtt{P}$ and $\mathtt{X}$, solve for Λ via least squares; (3) Alternate between the above two steps till convergence. Usually, to avoid possible trivial solutions (e.g., all depths being zero, or all but 4 columns of the depth matrix are zeros, etc.), some kind of row-wise and column-wise normalization (a.k.a *balancing*) is necessary.

2.2 Element-wise factorization

As we have explained earlier, though many of the existing iterative projective factorization algorithms do produce sufficiently good results, there is however, no any theoretic justification. In other words, the optimality of such iteration procedures is not guaranteed. In this subsection, we present a closed-form solution to projective factorization.

Our main idea. We repeat the basic equation here: $\mathtt{W} = \Lambda \odot \mathtt{M} = \mathtt{PX}$. Recall that $\mathtt{M}$ is the only input, and the task is to solve for both Λ and $\mathtt{W}$. Note that $\odot$ denotes element-wise product, therefore we can view the problem as an *element-wise factorization* problem, in the following sense:

– Given measurement matrix $\mathtt{M}$, find two matrices Λ and $\mathtt{W}$ such that $\mathtt{W} = \Lambda \odot \mathtt{M}$.

At a first glance, this seems to be an impossible task, as the system is severely under-constrained. However, for the particular problem of projective factorization, we have extra conditions on the unknown matrices which may sufficiently constrain the system. Roughly speaking, these extra conditions (to be listed below) are expected to supply the system with sufficient constraints, making the element-wise factorization problem well-posed and hence solvable. These constraints are in fact very mild, reasonable, and not restrictive.

– All visible points' projective depths must be positive. This is nothing but the well-known and very common *cheirality* constraint. In other words, visible point must lie in front of the camera.
– The re-scaled measurement matrix $\mathtt{W}$ has rank at most 4. This is true for noise-free case (we will further relax this in the actual computation).
– The rank of Λ is also at most 4. This is easy to see, since $[\lambda_{ij}]$ is a sub-matrix of $\mathtt{W}$; hence, $\mathrm{rank}(\mathtt{W}) \leq 4 \Rightarrow \mathrm{rank}(\Lambda) \leq 4$.
– All the rows and columns of Λ have been normalized to have (average) unit sum. The row-sum and column-sum constraints play two roles: (1) rule out trivial solutions;(2) rule out scale ambiguity in the factorization.

Formulation. Mathematically, the element-wise factorization is formulated as:

$$\begin{array}{l} \texttt{Find } \mathtt{W}, \Lambda, \texttt{ such that,} \\ \mathtt{W} = \Lambda \odot \mathtt{M}, \\ \mathrm{rank}(\mathtt{W}) \leq 4, \\ \sum_i \lambda_{ij} = m, j = 1, \cdots, n, \\ \sum_j \lambda_{ij} = n, i = 1, \cdots, m, \\ \lambda_{ij} > 0. \end{array} \tag{3}$$

There is one more theoretical issue left, however. One would ask: will the unit row-sum and column-sum constraints be too restrictive such that no feasible solution of Λ matrix can be found? This is a reasonable question to ask, because in the traditional iterative projective factorization scenarios, Mahamud et al [7] and Oliensis et al [8] both showed that applying the row-wise and column-wise normalization during the iteration may hinder the convergence.

However, we show that this is not a problem at all for our algorithm, thank to Sinkhorn's famous theorem regarding *doubly stochastic matrix* [11]. A square nonnegative matrix is called doubly stochastic if the sum of the entries in each row and each column is equal to one. Sinkhorn proved the following theorem, which gives the diagonal equivalence between doubly stochastic matrix and any arbitrary positive matrix.

Theorem 1. *[11] Any strictly positive matrix $\mathtt{A}$ of order n can always be normalized into a doubly stochastic matrix by the following diagonal scaling, $\mathtt{D}_1\mathtt{A}\mathtt{D}_2$, where $\mathtt{D}_1$ and $\mathtt{D}_2$ are diagonal matrices of order n with strictly positive diagonal entries. Such two diagonal matrices are unique up to scale for a given positive matrix $\mathtt{A}$.*

This theorem can be naturally generalized to non-square positive matrices, and we have the following result:

Corollary 1. *[12] Any strictly positive matrix $\mathtt{A}$ of size $m \times n$ can always be rescaled to $\mathtt{D}_1\mathtt{A}\mathtt{D}_2$ whose row-sums all equal to n and column-sums all equal to m, where $\mathtt{D}_1$ and $\mathtt{D}_2$ are respectively $m \times m$ and $n \times n$ positive diagonal matrices. Such $\mathtt{D}_1$ and $\mathtt{D}_2$ are unique up to scale for any given $\mathtt{A}$.*

In our context, this result suggests that the row-wise and column-wise normalization conditions are not restrictive, because the entire set of (positive) projective depth matrices is reachable from a row- and column-normalized positive matrix. Furthermore, in Appendix we will show that for general configurations, the rank = 4 constraint provides sufficient constraints for solving the problem.

3 Implementation

3.1 Rank Minimization

Noise is inevitable in real measurements, which will consequently increase the actual rank of $\mathtt{W}$. To accommodate noise, we slightly modify the problem formulation, and pose it as a rank minimization problem:

$$\begin{aligned}
&\texttt{Minimize}\ \operatorname{rank}(\mathtt{W}),\ \texttt{subject to},\\
&\mathtt{W} = \Lambda \odot \mathtt{M},\\
&\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n,\\
&\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m,\\
&\lambda_{ij} > 0.
\end{aligned} \tag{4}$$

Once the problem is solved, we can use Λ as the estimated depth matrix, and $\mathtt{W}$ as the rescaled measurement matrix. Subsequently they can be fed into a single SVD, or be used to initialize a bundle adjustment process.

3.2 Trace Minimization

To solve rank minimization problem exactly is intractable in general [13]. To overcome this, *nuclear-norm* has been introduced as the tightest convex surrogate of rank. The nuclear norm of $\mathtt{X} \in \mathbb{R}^{m\times n}$ is defined as $\|\mathtt{X}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i$, where σ_i is the ith singular value of $\mathtt{X}$.

Recently, using nuclear norm minimization to solve rank minimization problem has received considerable attention, in particular in the research of *compressive sensing*. One surprising result is that for a large class of matrices satisfying some "incoherency" or "restricted isometry" properties, nuclear norm minimization actually gives an exact solution. In other words, the relaxation gap is zero.

In this paper, we simply use the nuclear norm only as a convex surrogate (a relaxation) to the rank function, mainly for the purpose of approximately solving our projective factorization problem. Our main contribution of this work lies more in the new element-wise factorization formulation, than the actual computational implementation. We however appreciate the significance of the results due to compressive sensing. Thanks to these results, we at least can say, our algorithm may produce the exact and globally optimal solution, when certain conditions are satisfied.

Using the nuclear norm, we replace the original objective function $\mathrm{rank}(\mathtt{W})$ with $\|\mathtt{W}\|_*$. Furthermore, the nuclear norm minimization $\min \|\mathtt{W}\|_*$ can be rewritten as an equivalent SDP (semi-definite programming) problem:

$$\begin{aligned} &\min_{\mathtt{W}} \frac{1}{2}(\mathrm{tr}(\mathtt{X}) + \mathrm{tr}(\mathtt{Y})) \\ &\text{s.t.} \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0 \end{aligned}$$

Such an equivalence is grounded on the following theorem (ref. [14]).

Theorem 2. *Let* $\mathtt{A} \in \mathbb{R}^{m\times n}$ *be a given matrix, then* $\mathrm{rank}(\mathtt{A}) \leq r$ *if and only if there exist two symmetric matrices* $\mathtt{B} = \mathtt{B}^\top \in \mathbb{R}^{m\times m}$ *and* $\mathtt{C} = \mathtt{C}^\top \in \mathbb{R}^{n\times n}$ *such that* $\mathrm{rank}(\mathtt{B}) + \mathrm{rank}(\mathtt{C}) \leq 2r$ *and* $\begin{bmatrix} \mathtt{B} & \mathtt{A} \\ \mathtt{A}^\top & \mathtt{C} \end{bmatrix} \succeq 0$.

Piecing everything together, we finally reach a **trace minimization** problem:

$$\begin{aligned} &\min_{\mathtt{W}} \frac{1}{2}(\mathrm{tr}(\mathtt{X}) + \mathrm{tr}(\mathtt{Y})) \\ &\text{s.t.} \quad \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0 \\ &\mathtt{W} = \Lambda \odot \mathtt{M}, \\ &\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n, \\ &\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m, \\ &\lambda_{ij} > 0. \end{aligned} \tag{5}$$

This is a standard semi-definite programming (SDP), thus can be solved efficiently using any off-the-shelf SDP solvers. In all our experiments, we simply used SeDumi and SDPT3 [15] as the solvers, mainly for theory-validation purpose. Note however that, these state-of-the-art SDP solvers still cannot solve large scale problems, due to excessive memory and computational requirement. A better choice is those fast algorithms specially designed for large-scale nuclear norm minimization problems, and many of them can be found in recent compressive sensing literature (see e.g. [16],[17]).

4 Extensions

4.1 Dealing with missing data

In most real-world structure-from-motion applications, *missing data* are inevitable, due to e.g. self-occlusion, points behind cameras (i.e., cheirality) etc. Missing data lead to an incomplete measurement matrix, but simple SVD cannot directly perform on an incomplete matrix. This constitutes a major drawback of factorization-based methods.

For affine (camera) factorization, many missing data handling ideas have been proposed. Buchanan and Fitzgibbon [18] summarized existing methods, and classified them into four categories (1) closed form method,(2) imputation method, (3)alternation method and (4) direct nonlinear minimization.

Unfortunately, relatively less works were reported for projective factorization with missing data. A few related works are e.g. [19] [20] [21] [22]. Most existing works either rely on iteration or alternation, or assume the depths are precomputed by other means (reducing to affine case).

Our new element-wise factorization, on the other hand, offers a unified treatment to the missing data problem. With little modification, our SDP formulation can be extended to solve both complete case and missing data case. A similar work was reported elsewhere but is restricted to affine camera model [23].

Given an incomplete measurement matrix $\mathtt{M} = [\mathbf{m}_{ij}]$ with missing data, we define a 0-1 *mask matrix* Ω as

$$\Omega = [\omega_{ij}], \ \ \mathtt{where} \ \omega_{ij} = \begin{cases} \mathbf{1} \in \mathbb{R}^3, \ \mathtt{if} \ \mathbf{m}_{ij} \text{ is available,} \\ \mathbf{0} \in \mathbb{R}^3, \ \mathtt{if} \ \mathbf{m}_{ij} \text{ is missing.} \end{cases} \tag{6}$$

With these notations, the projective imaging process with missing data can be written as:

$$\Lambda \odot \mathtt{M} = \Omega \odot \mathtt{W}.$$

Now our task becomes:

- Given an incomplete $\mathtt{M}$, find a completed low-rank $\mathtt{W}$ such that $\Lambda \odot \mathtt{M} = \Omega \odot \mathtt{W}$.

Note that at those missing positions, we do not need to estimate the corresponding depths, so we set $\lambda_{ij} = 1$ whenever $\omega_{ij} = \mathbf{0}$.

Applying the nuclear norm heuristics, our SDP formulation for projective factorization with missing data is:

$$\begin{aligned} & \min_{\mathtt{W}} \frac{1}{2}(\mathrm{tr}(\mathtt{X}) + \mathrm{tr}(\mathtt{Y})) \\ & \text{s.t.} \ \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0, \\ & \Lambda \odot \mathtt{M} = \Omega \odot \mathtt{W}, \\ & \lambda_{ij} > 0, \ \text{if } \omega_{ij} = \mathbf{1}, \\ & \lambda_{ij} = 1, \ \text{if } \omega_{ij} = \mathbf{0}, \\ & \textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n, \\ & \textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m. \end{aligned} \tag{7}$$

Once this SDP converges, the resultant $\mathtt{W}$ is a completed $3m \times n$ full matrix with no entries missing. Moreover, we can even read out a *completed* projective depth matrix just as the sub-matrix of $\mathtt{W}$ formed by every the third rows of $\mathtt{W}$.

4.2 Dealing with pure outliers

Another recurring practical issue in real SFM applications is the outlier problem. Different from the missing data case, for the outlier case, we know that some of the entries of the given measurement matrix $\mathtt{M}$ are contaminated by gross errors (i.e., wrong matches), but we do not know where they are. We assume there are only a small portion of outliers and they are randomly distributed in $\mathtt{M}$, in other words, the outliers are sparse.

By conventional factorization methods, there is no easy and unified way to deal with outliers. Most published works are based on some pre-processing using RANSAC [19]. However, we show how our element-wise factorization formulation can handle this problem nicely and uniformly, if certain compressive sensing conditions are satisfied ([24],[25]).

Denote the actual measurement matrix as $\mathtt{M}$, which contains some outliers at unknown positions. Denote the underlying outlier-free measurement matrix as $\hat{\mathtt{M}}$. Then we have $\mathtt{M} = \hat{\mathtt{M}} + \mathtt{E}$, where $\mathtt{E}$ gives the outlier pattern. Now, list the basic projective imaging equation as $\mathtt{W} = \Lambda \odot (\mathtt{M} - \mathtt{E})$, the task is to simultaneously find the optimal $\mathtt{W}$,Λ and the outlier pattern $\mathtt{E}$, such that $\mathtt{W}$ has the lowest rank and $\mathtt{E}$ is as sparse as possible. To quantify the sparseness of $\mathtt{E}$, we use its element L_1-norm $\|\mathtt{E}\|_1 = \sum_{i,j} |\mathtt{E}_{i,j}|$ as a relaxation of its L_0-norm[5]. Combined with the nuclear-norm heuristics for $\mathtt{W}$, the objective function is chosen as $\|\mathtt{W}\|_* + \mu \|\mathtt{E}\|_1$, where μ is a trade-off parameter (we used 0.4 in our experiments).

The final minimization formulation for factorization with outliers becomes:

$$\begin{aligned}
&\min_{\mathtt{W},\mathtt{E}} \frac{1}{2}(\mathrm{tr}(\mathtt{X}) + \mathrm{tr}(\mathtt{Y})) + \mu \|\mathtt{E}\|_1 \\
&\text{s.t.} \quad \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0 \\
&\mathtt{W} = \Lambda \odot (\mathtt{M} - \mathtt{E}) \\
&\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n \\
&\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m \\
&\lambda_{ij} > 0, \forall i, j.
\end{aligned} \tag{8}$$

It is worth noting that, in the presence of missing data or outliers, the performance of our algorithm is problem-dependent. The ratio and the (spatial) distribution of outliers or missing data all affect the final result. But this is also true for most other algorithms.

5 Experimental Results

To evaluate the performance of the proposed method, we conducted extensive experiments on both synthetic data and real image data. We tested complete

[5] We use the fact that $\|\mathtt{E}\|_0 = \|\Lambda \odot \mathtt{E}\|_0$, since $\lambda_{ij} > 0$.

measurement case, as well as missing data case and outlier case. Reprojection error in the image plane and relative projective depth error (if the ground-truth is known) are used to evaluate the algorithm performance.

Relative depth error is defined as follows

$$\varepsilon = \frac{\|\hat{\Lambda}_{GT} - \hat{\Lambda}_{Recover}\|}{\|\hat{\Lambda}_{GT}\|}, \tag{9}$$

where $\hat{\Lambda}_{GT}$ is the ground truth projective depth matrix after column-sum and row-sum balancing and $\hat{\Lambda}_{Recover}$ is the projective depth matrix recovered after balancing.

5.1 Synthetic experiments

In all the synthetic experiments, we randomly generated 50 points within a cube of $[-30, 30]^3$ in space, while 10 perspective cameras were simulated. The image size is set as 800×800. The camera parameters are set as follows: the focal lengths are set randomly between 900 and 1100, the principal point is set at the image center, and the skew is zero. We added realistic Gaussian noise to all simulated measurements.

We first tested for the complete measurement case, i.e., the input measurement matrix $\mathtt{M}$ is complete. In all of our experiments, the SDP solver output results in less than 20 iterations (even including the experiments for missing data and outlier cases), and cost less than 0.5 seconds per iteration on a modest 1.6GHz Core-Duo laptop with memory 2GB using SDPT3 as solver.

Synthetic images: large depth variations. We simulated cases where the true depths are widely distributed and not close to one, which are commonly encountered in real world applications of structure from motion especially in large scale reconstructions.

We defined the depth variation as $r = \max_{ij}(\lambda_{ij})/\min_{ij}(\lambda_{ij})$, i.e. the ratio between the maximal depth and the minimal depth. We tested two cases, one is that all the depth variations are within [1,5), the other is that all the depth variations are within [5,20]. As we expect, our method outperforms all state-of-the-art iterative methods by a significant margin. Figure 1 illustrates error histograms for the two cases. From Fig. 1(a) and Fig. 1(c), we observe that our algorithm produces reprojection error less than 2 pixels while SIESTA (with balancing) outputs reprojection error up to 100 pixels for small and modest depth variations. From Fig. 1(b) and Fig. 1(d), we observe that our algorithm produces reprojection error less than 14 pixels while SIESTA outputs error up to 250 pixels for large depth variation. This can be explained that our method is a closed-form solution and does not depend on initialization. However all other algorithms highly depend on initialization, where affine camera model is widely used as initialization which is not the case for large depth variation.

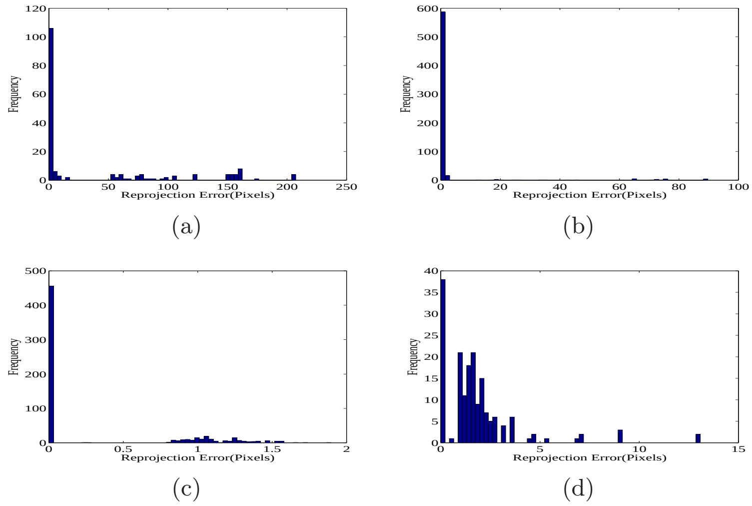


Fig. 1. Performance comparison between SIESTA and proposed method under various level of depth variations. Clearly, our method is much more superior. (a) Histogram of reprojection error by the normalized SIESTA ($r < 5$); (b) Histogram of reprojection error by the normalized SIESTA ($5 \leq r < 20$); (c) Histogram of reprojection error by our method ($r < 5$); (d) Histogram of reprojection error by our method ($5 \leq r < 20$).

Synthetic images: missing data. To evaluate the performance of our algorithm on measurements with missing data, we generated synthetic data sets with dimension 20×50 as before followed by removing 20% of 2D points in the measurement matrix randomly to simulate missing data case. The relative depth error and reprojection error for both visible points and missing points are plotted against different Gaussian noise levels in Fig. 2.

Synthetic images: outliers. To illustrate the performance of our algorithm for projective factorization with outliers, we generated the following illustrative example. The configuration is 10 cameras observing 20 points leading to measurement matrix of 20×20. The outlier pattern is generated according to uniform distribution with 5% positions are outliers. The results are shown in Fig. 3. From the figure, we conclude that our method recovers the outlier pattern successfully.

5.2 Real image experiments

Real images: complete data. We first tested our method on real images with complete measurement. Some of the real images used in our experiments are shown in Fig. 4. Reprojection errors for these images are shown in Table 1.

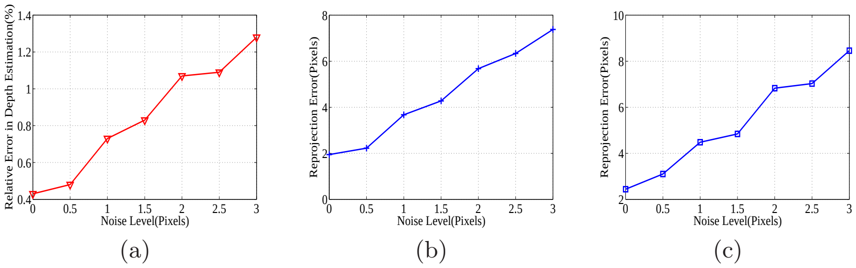


Fig. 2. Performance of the proposed method for missing data case. (a) Relative error in depth estimation under various level of Gaussian noise. (b) Reprojection error on visible points; (c) Reprojection error on all points.

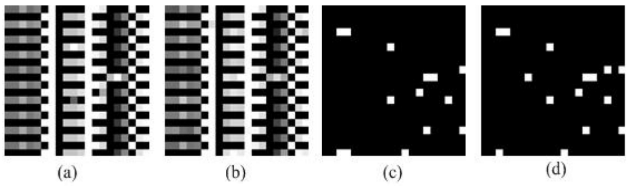


Fig. 3. Performance of the proposed method dealing with outliers in measurement matrix. (a) Input Measurement Matrix. (b) Recovered Measurement Matrix. (c) Recovered Outlier Pattern. (d) Ground Truth Outlier Pattern where white denotes outlier.

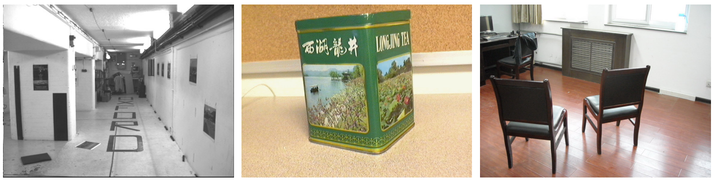
Real images: missing data. We tested our method on real images with missing data. A small portion of the Dinosaur data with dimension 18×20 is used, as our current SDP solver can only solve toy-size problems.

The Dinosaur sequence [18] is conventionally used as an example for affine factorization. Here we however solve it as a projective factorization with missing data problem. Our experiment is mainly for theory validation purpose. Fig. 5 illustrates the effect of our projective depth estimation at a 4×4 image patch.

6 Conclusion

In this paper, we have proposed a new element-wise factorization framework for non-iterative projective reconstruction. We formulate the problem as an SDP and solve it efficiently and (approximately) globally. Our results are comparable or superior to other iterative methods when these methods work. When they no longer work, ours still works.

Future work will address drawbacks of the current implementation, in particular the scalability issue of the standard SDP solver. We will also consider non-rigid deformable motion, degenerate cases, and cases combining missing data



(a) (b) (c)

Fig. 4. Real image sequences used in experiments. (a) Corridor, (b) Teabox, (c) Chair.

Table 1. Performance Evaluation for Real Images with Complete Measurements

Dataset / Method	SIESTA[8]	CIESTA[8]	Col-space[7]	Our method
Corridor	0.3961	0.3955	0.3973	0.3961
Teabox	4.4819e-4	4.4872e-4	4.8359e-4	4.4819e-4
Chair	1.3575	1.3723	1.3441	1.3385

$$\boxed{\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}} \qquad \boxed{\begin{array}{cccc} 1.0007 & 1.0011 & 0.9999 & 0.9958 \\ 0.9942 & 0.9934 & 0.9987 & 0.9878 \\ 0.9856 & 0.9837 & 0.9998 & 0.9808 \\ 0.9776 & 0.9749 & 1.0014 & 0.9776 \end{array}}$$

(a) (b)

Fig. 5. Depth estimation from affine factorization and projective factorization. (a) Affine camera sets all the depths to be 1s (b) Depths estimated by our method.

and outliers. Theoretical analysis about our missing-data and outlier-handling procedure is also planned.

Acknowledgements

The first author appreciated the China Scholarship Council for supporting his visit to ANU (Oct 2008 to Oct 2009). The authors would like to thank Prof. Richard Hartley for his immeasurable discussion. This work was supported partially by National Natural Science Foundation of China under key project 60736007, Natural Science Foundation of Shaanxi Province under 2010JZ011, ARC through its SRI in Bionic Vision Science and Technology grant to Bionic Vision Australia. NICTA is funded by the Australian Government as represented by the DBCDE and by the ARC.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vision **9** (1992) 137–154

2. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: ECCV, Springer-Verlag (1996) 709–720
3. Triggs, B.: Factorization methods for projective structure and motion. In: CVPR, IEEE Computer Society (1996) 845–851
4. Ueshiba, T., Tomita, F.: A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In: ECCV. (1998) 296–310
5. Heyden, A., Berthilsson, R., Sparr, G.: An iterative factorization method for projective structure and motion from image sequences. Image Vision Comput. **17** (1999) 981–991
6. Chen, Q., Medioni, G.: Efficient iterative solution to m-view projective reconstruction problem. CVPR (1999) 55–61
7. Mahamud, S., Hebert, M.: Iterative projective reconstruction from multiple views. In: CVPR. (2000) 430–437
8. Oliensis, J., Hartley, R.: Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence. IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 2217–2233
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2004)
10. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably-convergent iterative methods for projective structure from motion. In: CVPR. (2001) 1018–1025
11. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann. Math. Statist. (1964) 876–879
12. Sinkhorn, R.: Diagonal equivalence to matrices with prescribed row and column sums. The American Mathematical Monthly (1967) 402–405
13. Recht, B., Fazel, M., Parrilo, P.: Guarranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. Technical report, California Institute of Technology (2007)
14. Fazel, M., Hindi, H., Boyd, S.: A rank minimization heuristic with application to minimum order system approximation. In: Proc. Am. Control Conf,. (2001) 4734–4739
15. Toh, K., Todd, M., Tutuncu, R.: Sdpt3 — a matlab software package for semidefinite programming. Optimization Methods and Software (1999) 545–581
16. Cai, J.F., Candes, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. http://arxiv.org/abs/0810.3286 (2008)
17. Ma, S., Goldfarb, D., Chen, L.: Fixed point and bregman iterative methods for matrix rank minimization. Math. Program., Ser. A (2009)
18. Buchanan, A.M., Fitzgibbon, A.W.: Damped newton algorithms for matrix factorization with missing data. In: CVPR. (2005) 316–322
19. Martinec, D., Pajdla, T.: Structure from many perspective images with occlusions. In: ECCV. (2002) 355–369
20. Tang, W., Hung, Y.: A subspace method for projective reconstruction from multiple images with missing data. Image Vision Comput. **24** (2006) 515 – 524
21. Martinec, D., Pajdla, T.: 3D reconstruction by fitting low-rank matrices with missing data. In: CVPR. (2005) 198–205
22. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR. (2007) 1–8
23. Olsson, C., Oskarsson, M.: A convex approach to low rank matrix approximation with missing data. In: SCIA. (2009) 301–309
24. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? http://arxiv.org/abs/0912.3599 (2009)

25. Chandrasekaran, V., Sanghavi, S., Parrilo, P., Willsky, A.: Rank-sparsity incoherence for matrix decomposition. http://arxiv.org/abs/0906.2220v1 (2009)

Appendix:

In this appendix, we will show that our formulation (i.e., (5)) is well-posed, meaning that any true solution is indeed the exact unique solution of the formulation. We assume the cameras and points are generically configured, image measurements are complete and noise-free—hence the rank is identically 4.

In the main body of our paper (subsection-2.2), we have already shown that applying the row- and column- normalization places no restriction to the solution space, given that all (visible) projective depths are positive.

Next, we need to show that enforcing the two rank conditions on both $\mathtt{W}$ and Λ provides sufficient constraints for solving the element-wise factorization problem.

Denote the image coordinate as $\mathbf{m}_{ij} = [u_{ij}, v_{ij}, 1]^T$, the projective depth as λ_{ij}, according to the rank-4 constraint on projective depth matrix Λ, the jth column of Λ is expressed as

$$\Lambda_j = a_{1j}\Lambda_1 + a_{2j}\Lambda_2 + a_{3j}\Lambda_3 + a_{4j}\Lambda_4.$$

Since $\mathtt{W}$ is expected to have rank 4 in general case, we have

$$\mathtt{W}_j = b_{1j}\mathtt{W}_1 + b_{2j}\mathtt{W}_2 + b_{3j}\mathtt{W}_3 + b_{4j}\mathtt{W}_4, \tag{10}$$

where $\mathtt{W}_j, j = 5, \cdots, n$ denotes the $j$the column of $\mathtt{W}$.

Substitute the image coordinates into the above equations, we have

$$\begin{aligned}
u_{ij}(a_{1j}\lambda_{i1} + a_{2j}\lambda_{i2} + a_{3j}\lambda_{i3} + a_{4j}\lambda_{i4}) &= b_{1j}u_{i1}\lambda_{i1} + b_{2j}u_{i2}\lambda_{i2} + b_{3j}u_{i3}\lambda_{i3} + b_{4j}u_{i4}\lambda_{i4} \\
v_{ij}(a_{1j}\lambda_{i1} + a_{2j}\lambda_{i2} + a_{3j}\lambda_{i3} + a_{4j}\lambda_{i4}) &= b_{1j}v_{i1}\lambda_{i1} + b_{2j}v_{i2}\lambda_{i2} + b_{3j}v_{i3}\lambda_{i3} + b_{4j}v_{i4}\lambda_{i4} \\
a_{1j}\lambda_{i1} + a_{2j}\lambda_{i2} + a_{3j}\lambda_{i3} + a_{4j}\lambda_{i4} &= b_{1j}\lambda_{i1} + b_{2j}\lambda_{i2} + b_{3j}\lambda_{i3} + b_{4j}\lambda_{i4},
\end{aligned}$$

which implies that $a_{1j} = b_{1j}, a_{2j} = b_{2j}, a_{3j} = b_{3j}, a_{4j} = b_{4j}$. This can be explained as the rank-4 constraint on Λ is included under the rank-4 constraint on $\mathtt{W}$.

Then we have

$$\frac{u_{ij}}{v_{ij}} = \frac{b_{1j}\lambda_{i1}u_{i1} + b_{2j}\lambda_{i2}u_{i2} + b_{3j}\lambda_{i3}u_{i3} + b_{4j}\lambda_{i4}u_{i4}}{b_{1j}\lambda_{i1}v_{i1} + b_{2j}\lambda_{i2}v_{i2} + b_{3j}\lambda_{i3}v_{i3} + b_{4j}\lambda_{i4}v_{i4}}$$

Let $\eta_{ij} = \frac{u_{ij}}{v_{ij}}$, we obtain

$$\lambda_{i1}b_{1j}(u_{i1} - \eta_{ij}v_{i1}) + \lambda_{i2}b_{2j}(u_{i2} - \eta_{ij}v_{i2}) + \lambda_{i3}b_{3j}(u_{i3} - \eta_{ij}v_{i3}) + \lambda_{i4}b_{4j}(u_{i4} - \eta_{ij}v_{i4}) = 0$$

There are $m(n-4)$ equations while the number of variables is $4m+4(n-4)$, thus the problem is well-posed. In our SDP implementation, we use “min (rank)” (as opposed to enforcing a hard constraint of “rank=4”) to solve a relaxed version.