A Simple Prior-free Method for Non-Rigid Structure-from-Motion Factorization

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

Northwestern Polytechnical University, China[1]
Australian National University, Australia [2]

Abstract

This paper proposes a simple "prior-free" method for solving non-rigid structure-from-motion factorization problems. Other than using the basic low-rank condition, our method does not assume any extra prior knowledge about the nonrigid scene or about the camera motions. Yet, it runs reliably, produces optimal result, and does not suffer from the inherent basis-ambiguity issue which plagued many conventional nonrigid factorization techniques.

Our method is easy to implement, which involves solving no more than an SDP (semi-definite programming) of small and fixed size, a linear Least-Squares or trace-norm minimization. Extensive experiments have demonstrated that it outperforms most of the existing linear methods of nonrigid factorization. This paper offers not only new theoretical insight, but also a practical, everyday solution*, to non-rigid structure-from-motion.*

1. Introduction

This paper revisits the classical geometric computer vision problem of non-rigid structure-from-motion (NRSFM). We focus on the *factorization* framework for NRSFM, originally proposed by Bregler *et al*. in [7], as an important extension to the well-known Tomasi-Kanade factorization from rigid scene to nonrigid scene, assuming that the nonrigid shape deformation follows a low-order linear combination model. To date, a large body of researches has been devoted to this topic, and numerous different methods/algorithms have been proposed. However, despite all the efforts, this problem remains a difficult and still active research topic (see *e.g*., [5, 11, 25, 21, 3, 14]).

One of the primary causes to such difficulty is due to the inherent *basis ambiguity* of the nonrigid problem ([26]). To overcome this, most existing work rely on introducing various *prior* knowledge to the problem at hand. They do so by assuming various constraints about the nonrigid scene, about the nonrigid shape bases, about the coefficients, about the deformation, about the shape itself, or about the camera motion *etc*. For instance, many methods require that the camera moves smoothly, or the deformation trajectory is slow and smooth. However, these additional constraints not only limit the practical applicability of the methods, but also obscure a clear theoretical understanding to the problem. We would like to answer: in order to solve the nonrigid factorization effectively, are these additional priors *essential* ?

In this paper, we propose a novel and simple solution to non-rigid factorization. Our method does not assume any extra prior knowledge about the problem other than the low-rank constraint, hence it is "prior-free". Nevertheless, it does not suffer from the basis ambiguity difficulty, but is able to recover both camera motion and non-rigid shape accurately and reliably. Experiments on both synthetic and real benchmark datasets show that: the proposed method, being prior-free, outperforms most other (often prior-based) linear factorization methods.

To better present this paper, and also to put our contributions in context, we briefly review some recent progress in non-rigid factorization.

1.1. Related work

Ever since Bregler's seminal work [7] in 2000, researchers have been actively applying the factorization framework to various nonrigid problems. However, they soon noticed that, different from its rigid counterpart, the non-rigid factorization appeared to be much more difficult. In a 2004 paper, Xiao *et al*. proved that the problem itself is indeed ill-posed or under-constrained, in the sense that, based on the orthonormality constraint alone, one cannot recover the non-rigid shape bases and the corresponding shape coefficients *uniquely* [26]. There is always a fundamental ambiguity between the shape bases and the shape coefficients.

To resolve this ambiguity, Xiao *et al*. suggested to add extraneous "basis constraints" so as to make the system well-constrained. In the same spirit of adding extra priors to regularize an otherwise under-constrained problem, Torresani *et al*. [25] introduced Gaussian prior on the shape coefficients. Del Bue introduced special shape priors [11]. Akhter *et al*. proposed to use a fixed set of DCT bases in

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

the trajectory dual space [3]. Temporally smooth deformation prior has also been used such as in [5][1]. Gatardo *et al*. assumed the time-trajectory of a single point is smooth [14]. Other priors imposed speciality on the model such as assuming a quadratic model [13], local-rigidity [24], or extend to non-linear models [22].

Akhter *et al*. made an important theoretical progress in [2] which reveals that: although the ambiguity in shape basis is inherent, the 3D shape itself can be recovered uniquely without ambiguity. In a slightly earlier paper, Hartley and Vidal proved a similar result but under perspective camera model [16]. Despite the significance of these theoretical results, neither paper has provided a practical algorithm to the problem, and our work aims to fill this gap.

2. Problem Statement

2.1. Formulation

The task of nonrigid factorization is to factorize an *image measurement matrix* $\mathtt{W}$ as the product of *camera motion* (projection) matrix $\mathtt{M}$ and a nonrigid shape matrix $\mathtt{S}$, such that $\mathtt{W} = \mathtt{MS}$. We assume the measurement matrix is already centralized, therefore the camera matrix reduces to pure rotation [7].

Based on the linear combination model, the non-rigid shape $\mathtt{S}_i \in \mathbb{R}^{3\times P}$ can be represented as a linear combination of K *shape bases* $\mathtt{B}_k \in \mathbb{R}^{3\times P}$ with *shape coefficients* c_{ik} as: $\mathtt{S}_i = \sum_{k=1}^{K} c_{ik}\mathtt{B}_k$. Under orthographic camera model, the coordinates of the 2D image points observed at frame i are given by: $\mathtt{W}_i = \mathtt{R}_i\mathtt{S}_i$, where $\mathtt{R}_i \in \mathbb{R}^{2\times 3}$ is the first two rows of the i-th camera rotation, hence $\mathtt{R}_i\mathtt{R}_i^T = \mathtt{I}_2$. Use this representation, and stack all the F frames of measurements and all the P points in a matrix form, we obtain:

$$\begin{aligned}
\mathtt{W} &= \begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1F} \\ \vdots & & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathtt{R}_1\mathtt{S}_1 \\ \vdots \\ \mathtt{R}_F\mathtt{S}_F \end{bmatrix} \\
&= \begin{bmatrix} c_{11}\mathtt{R}_1 & \cdots & c_{1K}\mathtt{R}_1 \\ \vdots & \ddots & \vdots \\ c_{F1}\mathtt{R}_F & \cdots & c_{FK}\mathtt{R}_F \end{bmatrix} \begin{bmatrix} \mathtt{B}_1 \\ \vdots \\ \mathtt{B}_K \end{bmatrix} \qquad (1) \\
&= \mathtt{R}(\mathtt{C}\otimes\mathtt{I}_3)\mathtt{B} \doteq \Pi\mathtt{B}.
\end{aligned}$$

In this formula, we call $\mathtt{R} = \mathrm{blkdiag}(\mathtt{R}_1, \cdots, \mathtt{R}_F) \in \mathbb{R}^{2F\times 3F}$ the camera motion (rotation) matrix. Since $\Pi \in \mathbb{R}^{2F\times 3K}$ and $\mathtt{B} \in \mathbb{R}^{3K\times P}$, it is easy to see: $\mathrm{rank}(\mathtt{W}) \leq \min(\mathrm{rank}(\Pi), \mathrm{rank}(\mathtt{B})) \leq 3K$. Additionally, the shape matrix $\mathtt{S} = (\mathtt{C}\otimes\mathtt{I}_3)\mathtt{B}$ is low rank too, as $\mathrm{rank}(\mathtt{S}) \leq \min(\mathrm{rank}(\mathtt{C}\otimes\mathtt{I}_3), \mathrm{rank}(\mathtt{B})) \leq 3K$.

2.2. Orthonormality Constraint

From a measurement matrix $\mathtt{W}$ one can compute its rank-$3K$ decomposition $\mathtt{W} = \hat{\Pi}\hat{\mathtt{B}}$ via SVD. However, this decomposition is not unique as any nonsingular matrix $\mathtt{G} \in \mathbb{R}^{3K\times 3K}$ can be inserted between $\hat{\Pi}$ and $\hat{\mathtt{B}}$ to obtain a new valid factorization as $\mathtt{W} = \hat{\Pi}\hat{\mathtt{B}} = \hat{\Pi}\mathtt{G}\mathtt{G}^{-1}\hat{\mathtt{B}} = \Pi\mathtt{B}$.

A particular matrix $\mathtt{G}$ that rectifies $\hat{\Pi}$ to be a canonical Euclidean form is called the *Euclidean corrective matrix*, because once such a $\mathtt{G}$ is determined, one obtains $\mathtt{R}(\mathtt{C}\otimes\mathtt{I}_3) = \hat{\Pi}\mathtt{G}$ and the true shape bases $\mathtt{B} = \mathtt{G}^{-1}\hat{\mathtt{B}}$.

Denote the i-th double rows of $\hat{\Pi}$ as $\hat{\Pi}_{2i-1:2i} \in \mathbb{R}^{2\times 3K}$, and the k-th column-triplet of $\mathtt{G}$ as $\mathtt{G}_k \in \mathbb{R}^{3K\times 3}$, we have:

$$\hat{\Pi}_{2i-1:2i}\mathtt{G}_k = c_{ik}\mathtt{R}_i, \;\; i = 1, \cdots, F, k = 1, \cdots, K. \qquad (2)$$

Orthonormality constraints (*i.e*. rotation constraints) in the Π matrix can be imposed to recover a Gram matrix $\mathtt{Q}_k \in \mathbb{R}^{3K\times 3K}$ formed by $\mathtt{Q}_k = \mathtt{G}_k\mathtt{G}_k^T$ as $\hat{\Pi}_{2i-1:2i}\mathtt{Q}_k\hat{\Pi}_{2i-1:2i}^T = c_{ik}^2\mathtt{I}_2$. Since c_{ik} is not known, one can only establish two linear equations over $\mathtt{Q}_k$ as:

$$\hat{\Pi}_{2i-1}\mathtt{Q}_k\hat{\Pi}_{2i-1}^T = \hat{\Pi}_{2i}\mathtt{Q}_k\hat{\Pi}_{2i}^T, \;\; \hat{\Pi}_{2i-1}\mathtt{Q}_k\hat{\Pi}_{2i}^T = 0. \qquad (3)$$

2.3. Inherent Ambiguity

In doing the above non-rigid factorization, Xiao *et al*. [26] discovered that, the solutions are however fundamentally ambiguous, in the sense that one cannot expect to find the shape bases and shape coefficients uniquely. Such an inherent ambiguity largely explains why nonrigid factorization is fundamentally more difficult than its rigid counterpart. Later, Akhter *et al*. [2] showed that, quite surprisingly, the fundamental ambiguity does not necessarily lead to an ambiguous shape. In addition, they further proved that using the orthonormality constraints alone is in fact sufficient to recover a unique (unambiguous) non-rigid shape (provided that a previously-overlooked rank-3 constraint on $\mathtt{Q}_k$ (Eq.-(3)) is accounted for). However, apart from its evident theoretical value, their paper did not propose any optimization algorithm (other than a local search method due to [6]) to efficiently find the correct $\mathtt{G}_k$. Instead, the authors argued that "the real difficulty of in achieving good 3D reconstructions for nonrigid structures...is not the ambiguity of the [basis] constraints, but the complexity of the underlying non-linear optimization". In this paper we will challenge this argument, by providing a simple yet efficient (optimization) solution to Nonrigid SFM Factorization.

3. Main Theory

From now on, let us assume the measurement matrix $\mathtt{W}$ is already truncated to rank $3K$ (by *e.g*. SVD), the number of shape bases K has been estimated, and all the shape bases are non-degenerate. We start with a known result of NRSFM factorization.

Theorem 3.1. *All the solutions of* $\mathtt{Q}_k$ *to linear system Eq.-(3) form a linear subspace of dimensionality* $(2K^2 - K)$.

This result is in fact a direct consequence of Xiao *et al.*'s Theorem in [26]. It shows that the above linear system is inherently under-determined (as by $(2K^2 - K)$ rank deficient), no matter how many image frames are given.

On the other hand, this result also provides us with the true dimensionality of the solution space of $\mathtt{Q}_k$, and note that $\mathtt{Q}_k$ is precisely what we are after. However, in their paper, this practical implication had not been explicitly exploited.

In the following, we will show how one can take advantage of this result, and derive a practical algorithm that directly leads to a parametrization of this solution space. More precisely, we will prove that the solution space of $\mathtt{Q}_k$ is actually the *null-space* of a certain matrix $\mathtt{A}$ which can be directly obtained from the input image data. Practical usefulness of this representation is obvious.

3.1. Null-space Representation

First, denote vec() as the vectorization operator, and $\mathbf{q}_k = \text{vec}(\mathtt{Q}_k)$. Using $\text{vec}(\mathtt{AXB}^T) = (\mathtt{B} \otimes \mathtt{A})\text{vec}(\mathtt{X})$, we rewrite the linear system Eq.-(3) as:

$$\begin{bmatrix} (\hat{\Pi}_i \otimes \hat{\Pi}_i)(1,:) - (\hat{\Pi}_i \otimes \hat{\Pi}_i)(4,:) \\ (\hat{\Pi}_i \otimes \hat{\Pi}_i)(2,:). \end{bmatrix} \mathbf{q}_k \doteq \mathtt{A}_i \mathbf{q}_k = \mathbf{0}, \tag{4}$$

where $(\hat{\Pi}_i \otimes \hat{\Pi}_i)(j,:)$ denotes the j-th row of $(\hat{\Pi}_i \otimes \hat{\Pi}_i)$.

Stacking all such equations from all frames ($i = 1, \ldots, F$), we then have

$$\mathtt{A}\text{vec}(\mathtt{Q}_k) = \mathtt{A}\mathbf{q}_k = \mathbf{0}, \tag{5}$$

where $\mathtt{A} = [\mathtt{A}_1^T, \mathtt{A}_2^T, \cdots, \mathtt{A}_F^T]^T$. This is a linear system of equations over the unknown $9K^2$-vector $\mathbf{q}_k$.

Note that the $9K^2$-vector $\mathbf{q}_k$ has $(3K)(3K+1)/2$ independent entries. It may appear that, given enough frames, *i.e.* when $2F \geq (3K)(3K+1)/2$, $\mathbf{q}_k$ should be able to be solved via linear least squares. However, this is not the case, because all valid solutions reside in a $2K^2 - K$ dimensional space as shown in Theorem 3.1. Moreover, Eq.-(5) shows, the solution space is nothing but the null-space of $\mathtt{A}$. In addition, it is easy to verify that the minimum required number of frames for computing the null-space linearly is $F \geq (5K^2 + 5K)/4$.

3.2. The Intersection Theorem

Combining all the proceeding results, we now arrive at the central theorem of this paper:

Theorem 3.2 (Intersection Theorem). *Under non-degenerate and noise-free conditions, any correct solution of* $\mathtt{Q}_k$ (i.e. *the Gram matrix of a column-triplet of the true Euclidean corrective matrix* $\mathtt{G}_k$*) must lie in the intersection of the* $(2K^2 - K)$*-dimensional null-space of* $\mathtt{A}$ *and a rank-3 positive semi-definite matrix cone,* i.e., $\mathtt{Q}_k$ *belongs to*

$$\{\mathtt{A}\ \text{vec}(\mathtt{Q}_k) = \mathbf{0}\} \cap \{\mathtt{Q}_k \succeq 0\} \cap \{\text{rank}(\mathtt{Q}_k) = 3\}. \tag{6}$$

Proof. Denote $\mathtt{G}_k$ as column triplet of a correct rectifying transform, and $\mathtt{Q}_k = \mathtt{G}_k\mathtt{G}_k^T$, then $\text{rank}(\mathtt{Q}_k) = \text{rank}(\mathtt{G}_k) = 3, \mathtt{Q}_k \succeq 0$. Additionally, $\text{vec}(\mathtt{Q}_k)$ lies in the null space of $\mathtt{A}$ as $\text{vec}(\mathtt{Q}_k)$ gives correct rectification with zero error. Thus $\mathtt{G}_k$ is a solution to the equation system which means that the equation system is well-defined. Denote $\tilde{\mathtt{G}}_k$ as solution to the equation system Eq.-(6), then $\text{rank}(\tilde{\mathtt{Q}}_k) = 3, \tilde{\mathtt{Q}}_k \succeq 0$ and $\text{vec}(\tilde{\mathtt{Q}}_k)$ lies in the null space of the system of linear constraints on the elements $\tilde{\mathtt{Q}}_k$. Thus all the solutions to the equation system Eq.-(6) satisfy the condition for correct rectifying transforms. There is no difference between these solutions. □

4. Algorithm Solution

Armed with the above results (in particular Theorem 3.2), we are now ready to present our simple algorithm to the non-rigid factorization problem.

Recall that the goal is to recover the true motion matrix $\mathtt{R}$ and the true non-rigid shape matrix $\mathtt{S}$ from image measurement $\mathtt{W}$, such that $\mathtt{W} = \mathtt{RS} = \mathtt{R}(\mathtt{C} \otimes \mathtt{I}_3)\mathtt{B}$. Note that due to the inherent basis ambiguity, it is hopeless to recover a unique $\mathtt{B}$ or $\mathtt{C}$. While in previous work many researchers chose to use a pre-selected special shape bases $\mathtt{B}$ (or enforce arbitrary priors on the shape bases or shape coefficients) to pin down the undetermined degrees-of-freedom, in this work we will show how one can directly estimate the $\mathtt{S}$ without fixing $\mathtt{B}$ or $\mathtt{C}$.

Our algorithm consists of three steps to be applied in sequel: (1) Estimate the (Gram of) corrective matrix $\mathtt{G}_k$, (2) Estimate camera rotations $\mathtt{R}$ and (3) Estimate the nonrigid shape $\mathtt{S}$. We now explain the three steps in order.

4.1. Step-1: Estimate $\mathtt{G}_k$ by Trace-Minimization

Our main intersection theorem (Theorem-3.2) naturally leads to an easy algorithm to solve for $\mathtt{G}_k$, that is: to find the intersection of the aforementioned null-space and a rank-3 positive semi-definite matrix cone.

Because the rank-function itself is not very numerically-stably, measurements noise will increase the *numerical rank* of $\mathtt{Q}_k$ dramatically, we slightly relax the $\text{rank}(\mathtt{Q}_k) = 3$ condition to a *rank-minimization* problem, *i.e.* $\min \text{rank}(\mathtt{Q}_k)$. Note that however, rank-minimization is an NP-hard problem in general, and is very difficult to solve exactly. We therefore further relax it to a nuclear-norm minimization form, *i.e.*, $\min \|\mathtt{Q}_k\|_*$. Moreover in our case, since $\mathtt{Q}_k$ is a symmetric positive definite matrix, the nuclear norm is simply its trace [23][4]. Thus we have $\|\mathtt{Q}_k\|_* = \text{trace}(\mathtt{Q}_k)$.

Then we arrive at the following trace-minimization to solve for the corrective matrix.

$$\begin{aligned} &\min \text{ trace}(\mathtt{Q}_k), \text{ such that,} \\ &\mathtt{Q}_k \succeq 0, \\ &\mathtt{A}\ \text{vec}(\mathtt{Q}_k) = \mathbf{0}. \end{aligned} \tag{7}$$

To avoid a trivial solution at $\mathtt{Q}_k = \mathbf{0}$, we express $\mathrm{vec}(\mathtt{Q}_k)$ in explicit form of the null-space representation.

Easy to see that the above trace-minimization problem is a standard semi-definite programming (SDP). Also note that this SDP is actually of small and fixed size (of $2K^2 - K$) which is independent of the size of the measurement matrix. Thus this SDP can be solved very easily and very efficiently by any off-the-shelf SDP solvers. Once $\mathtt{Q}_k$ is found, we use SVD to extract $\mathtt{G}_k$. This solved $\mathtt{G}_k$ can be directly used to find $\mathtt{R}$ and then $\mathtt{S}$. Alternatively, if higher accuracy is desired, one can further improve the numerical accuracy of $\mathtt{G}_k$ by feeding it as an initial point to a non-linear refinement procedure, such as via the following unconstrained minimization: $\min_{\mathtt{G}_k} \sum_{i=1}^{F} \left[(1 - \frac{\hat{\Pi}_{2i}\mathtt{G}_k\mathtt{G}_k^T\hat{\Pi}_{2i}^T}{\hat{\Pi}_{2i-1}\mathtt{G}_k\mathtt{G}_k^T\hat{\Pi}_{2i-1}^T})^2 + (2\frac{\hat{\Pi}_{2i-1}\mathtt{G}_k\mathtt{G}_k^T\hat{\Pi}_{2i}^T}{\hat{\Pi}_{2i-1}\mathtt{G}_k\mathtt{G}_k^T\hat{\Pi}_{2i-1}^T})^2 \right]$, where the objective function is nothing but the orthonormality condition and this refinement is similar to a bundle-adjustment process.

4.2. Step-2: Compute Rotation Matrix $\mathtt{R}$

Conventionally, once $\mathtt{G}_k \in \mathbb{R}^{3K\times 3}$ is solved (w.l.o.g., let's denote it as $\mathtt{G}_1$), which is merely a single column-triplet in the full corrective matrix $\mathtt{G} \in \mathbb{R}^{3K\times 3K}$, the commonly-used next step is to solve for the other $K-1$ of independent column-triplets $[\mathtt{G}_2, \ldots, \mathtt{G}_K]$, and use them to populate the entire matrix $\mathtt{G}$. Brand [6] proposed a linear method to solve for the big $\mathtt{G}$ (for affine case). Because these $\mathtt{G}_k$s always have rotation ambiguity, in order to align them, Procrustes method must be employed subsequently (c.f. [26][2]). Once the big $\mathtt{G}$ is obtained, one then is allowed to compute the camera motion $\mathtt{R}$, the shape coefficients $\mathtt{C}$ and the shape bases $\mathtt{B}$, and then reconstruct the non-rigid shape $\mathtt{S}$. However, the above approach is not only rather involved, but also not numerically stable. More importantly, it is *not necessary*, as shown in [3].

In this work, we adopt a simpler approach that directly computes the camera motion $\mathtt{R}$ from a single column-triplet $\mathtt{G}_k$, without the need to fill in a big and full $\mathtt{G}$ matrix. The method goes as follows. Once $\mathtt{G}_k$ is solved, the rotation at every frame $i = 1, \cdots, F$ can be solved by using:

$$\hat{\Pi}_{2i-1:2i}\mathtt{G}_k = c_{ik}\mathtt{R}_i, \; i = 1, \cdots, F. \tag{8}$$

Note that we do not need to care about the unknown value of c_{ik}, though its sign ambiguity must be taken care of (c.f. [2]). Finally, the full motion matrix $\mathtt{R}$ is formed as $\mathtt{R} = \mathrm{blkdiag}([\mathtt{R}_1, \mathtt{R}_2, \ldots, \mathtt{R}_F])$.

4.3. Step-3: Estimate $\mathtt{S}$ by Rank-Minimization

Now we show how to solve the non-rigid shape matrix $\mathtt{S}$. Most conventional methods do this *indirectly*, in the sense that they often start from solving the big $\mathtt{G}$ matrix firstly, and then use pre-selected special shape bases $\mathtt{B}$ (such as the first K frame [26], or DCT bases in the dual space [3], or assume the the shape coefficients are also DCT-expandable [14]), then the corresponding coefficient matrix $\mathtt{C}$ can be determined, and also the shape matrix $\mathtt{S} = (\mathtt{C} \otimes \mathtt{I}_3)\mathtt{B}$.

In the next two sub-sections, we will provide two simpler, more direct methods for solving $\mathtt{S}$.

4.3.1 Pseudo Inverse Method

Recall that our goal is to solve $\mathtt{S}$ through the equation of $\mathtt{W} = \mathtt{RS}$ given $\mathtt{W}$ and $\mathtt{R}$. This equation is under-determined because $\mathtt{R}$ is a short matrix of size $2F \times 3F$. There should be no unique but an infinite family of solutions to $\mathtt{S}$. However, we also notice that: the low-order linear model, *i.e.* $\mathtt{S} = (\mathtt{C} \otimes \mathtt{I}_3)\mathtt{B}$ immediately suggests that $\mathrm{rank}(\mathtt{S}) \leq 3K$.

Taking into account of both of the above arguments, we reach: a valid solution to the shape matrix $\mathtt{S}$ must lie in the intersection of low-rank matrix set of $\{\mathrm{rank}(\mathtt{S}) \leq 3K\}$ and the solution space of equation $\mathtt{W} = \mathtt{RS}$. As usual we relax the low-rank condition to rank-minimization. Now the shape matrix $\mathtt{S}$ must be a solution to the following rank minimization problem:

$$\begin{aligned} &\min \mathrm{rank}(\mathtt{S}), \mathrm{such\ \ that}, \\ &\mathtt{W} = \mathtt{RS}. \end{aligned} \tag{9}$$

Remarks. We now make two important remarks: (1) the above rank-minimization problem (in the context of NRSFM) must have a unique solution; (2) this unique solution is nothing but the (unique) Moore-Penrose pseudo-inverse solution, *i.e.* $\mathtt{S} = \mathtt{R}^{\dagger}\mathtt{W} = (\mathtt{R}^T(\mathtt{R}\mathtt{R}^T)^{-1})\mathtt{W}$.

Remark-1 is not surprising, as it is simply the main conclusion of [2], which states that: once $\mathtt{R}$ is fixed, there is no ambiguity in finding the shape matrix $\mathtt{S}$, and the solution is *unique*.

Remark-2 was a bit surprising to the authors, as it seems to suggest a simple, linear, and closed-form solution to our rank-minimization problem of (9)–which otherwise would be NP-hard to solve in general.

Fortunately, recent progress in Compressive Sensing has confirmed the correctness of our Remark-2. In particular, we use the following result due to [19] that: *the Moore-Penrose pseudo-inverse solution* $\mathtt{S} = \mathtt{R}^{\dagger}\mathtt{W}$ *is the* unique minimizer *of the above rank minimization problem* (9). For a detailed proof the reader is referred to [19]. From this result, we see that the pseudo-inverse solution is indeed a *unique* matrix that satisfies the necessary conditions which a low-rank shape matrix must all satisfy.

We have applied the pseudo-inverse method to both synthetic data and real data. In terms of shape-recovery accuracy, the pseudo-inverse solution outperforms Xiao *et al.*'s K-basis method by a large margin, and achieves comparable performance with Metric-projection [21] and EM-PPCA [25]. The reader is referred to the second last column of

Table-1. This is already very encouraging, as our method does not use any priors except for the low-rank condition. Note that however, our method is slightly inferior to the more recent DCT trajectory basis method [3] or the CS-F method [14] (, both methods rely on strong smoothness prior)–which prompts us to think: can we do any better, without using any prior ?

4.3.2 Block Matrix Method

In the above pseudo-inverse method, we mainly make use of the rank-3K condition that is rank$(\mathtt{S}) \leq 3K$. This $(3F \times P)$ matrix $\mathtt{S}$ is simply a stack of P 3D points $[X_i, Y_i, Z_i]^T$ over F frames.

However, we realize that, since in reality there are *in fact* only K shape bases (rather than $3K$), the shape matrix $\mathtt{S}$ is not a fully-generic rank-$3K$ matrix, but has its special block structure.

In particular, we re-arrange the rows of $\mathtt{S}$ that correspond to X, Y,and Z coordinate separately, in an $F \times 3P$ block matrix form, denoted by $\mathtt{S}^\sharp$ in below[1]:

$$\mathtt{S}^\sharp = \begin{bmatrix} X_{11} & .. & X_{1P} & Y_{11} & .. & Y_{1P} & Z_{11} & .. & Z_{1P} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ X_{F1} & .. & X_{FP} & Y_{F1} & .. & Y_{FP} & Z_{F1} & .. & Z_{FP} \end{bmatrix}.$$

Then we must have: rank$(\mathtt{S}^\sharp) \leq K$. Note that this rank-$K$ condition (on $\mathtt{S}^\sharp$) is stronger than the above rank-$3K$ condition (on $\mathtt{S}$), and the former captures the essence of the K-order linear combination model.

Now, to solve for this re-arranged shape matrix $\mathtt{S}^\sharp$, we use, again, a rank-minimization formulation:

$$\begin{aligned} &\min \text{rank}(\mathtt{S}^\sharp), \text{such that}, \\ &\mathtt{W} = \mathtt{R}\mathtt{S}, \\ &\mathtt{S}^\sharp = [\mathtt{P}_X \ \mathtt{P}_Y \ \mathtt{P}_Z](\mathtt{I}_3 \otimes \mathtt{S}). \end{aligned} \tag{10}$$

The last matrix equality condition is a compact (shorthand) representation of the re-arrangement relationship between $\mathtt{S}^\sharp$ and $\mathtt{S}$, where $\mathtt{P}_X, \mathtt{P}_Y, \mathtt{P}_Z \in \mathbb{R}^{F \times 3F}$ are some properly defined 0-1-valued "row-selection" matrices (similar to the "permutation matrix").

Fast numerical implementation. We relax the above rank-minimization to nuclear-norm (*i.e.* trace-norm) minimization, *i.e.* $\min \|\mathtt{S}^\sharp\|_*$. In principle, this nuclear-norm minimization may be solved by a standard SDP solver. However, unlike the case of Eq.-7 where the resulted SDP has small and fixed size, here this SDP is of size $F \times 3P$, which renders the SDP technique very inefficient when either P or F is large.

Below, we give an efficient numerical implementation, based on *fixed point continuation* [20]. First, we re-cast the above minimization Eq.-(10) in Lagrangian form as:

$$\begin{aligned} &\min \mu \|\mathtt{S}^\sharp\|_* + \frac{1}{2}\|\mathtt{W} - \mathtt{R}\mathtt{S}\|_{\mathrm{F}}^2, \text{such that}, \\ &\mathtt{S}^\sharp = [\mathtt{P}_X \ \mathtt{P}_Y \ \mathtt{P}_Z](\mathtt{I}_3 \otimes \mathtt{S}), \end{aligned} \tag{11}$$

where μ is the continuation (homotopy) parameter which diminishes as the algorithm iterates. Next, the gradient of $\frac{1}{2}\|\mathtt{W} - \mathtt{R}\mathtt{S}\|_{\mathrm{F}}^2$ with respect to $\mathtt{S}^\sharp$ is obtained as:

$$g(\mathtt{S}^\sharp) = \frac{\partial \frac{1}{2}\|\mathtt{W} - \mathtt{R}\mathtt{S}\|_{\mathrm{F}}^2}{\partial \mathtt{S}^\sharp} = [\mathtt{P}_X \ \mathtt{P}_Y \ \mathtt{P}_Z](\mathtt{I}_3 \otimes (\mathtt{R}^T(\mathtt{R}\mathtt{S} - \mathtt{W}))). \tag{12}$$

Then, we solve the minimization of Eq.-(11) via the following two-line iteration update (cf. [20]):

$$\begin{cases} \mathtt{Y}^{(t)} = \mathtt{S}^{\sharp(t)} - \tau g(\mathtt{S}^{\sharp(t)}), \\ \mathtt{S}^{\sharp(t+1)} = \mathcal{S}_{\tau\mu}(\mathtt{Y}^{(t)}), \end{cases} \tag{13}$$

where τ is the step size of gradient descent, and $\mathcal{S}_v(\cdot)$ is the *matrix shrinkage* operator (cf. [20]). Once the iteration converges, we first project the solved $\mathtt{S}^\sharp$ to the nearest rank-K matrix (note: not $3K$), then rearrange it to $\mathtt{S}$.

5. Experiments

5.1. Setup

We compare our methods against the state-of-the-art methods, which include (1) Xiao *et al.*'s shape basis method (XCK) [26];(2) Torresani *et al.*'s EM-PPCA [25]; (3) Metric projection [21]; (4) Trajectory basis method [3]; and (5) Column space fitting (CSF) [14]. [2]

To facilitate the comparison, we use the same error metrics as reported in [3] and [14], that is: e_R measures the mean error in rotation estimation and $e_R = \frac{1}{F}\sum_{i=1}^{F} \|\mathtt{R}_i - \tilde{\mathtt{R}}_i\|_{\mathrm{F}}$, where $\mathtt{R}_i$ is the ground truth rotation at frame i and $\tilde{\mathtt{R}}_i$ the recovered rotation; e_{3D} measures the normalized mean 3D error in the reconstructed 3D points and $e_{3D} = \frac{1}{\sigma FP}\sum_{i=1}^{F}\sum_{p=1}^{P} e_{ip}$, $\sigma = \frac{1}{3F}\sum_{i=1}^{F}(\sigma_{ix} + \sigma_{iy} + \sigma_{iz})$, where σ_{ix}, σ_{iy} and σ_{iz} are the standard deviations in X, Y and Z coordinates of the original shape at frame i.

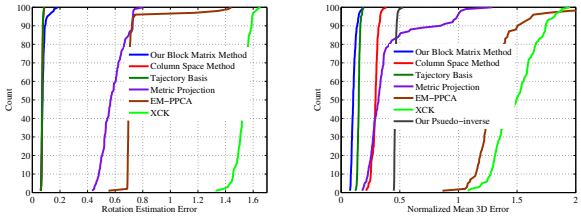
Extensive experiments are conduced to test the performance of the proposed methods, on both randomly synthetic data and on real motion capture data. The random synthetic data, which satisfy the low-rank nonrigid model perfectly, are used only for the purpose of algorithm validation, for which our methods have obtained nearly perfect result (with zero error) as we expect; the results are therefore omitted. Instead, only results on real sequences are reported below. The real sequences we have tested include the standard sequences of Drink $(1102/41)$, Pickup $(357/41)$, Yoga $(307/41)$, Stretch $(370/41)$, and Dance $(264/75)$ used in [3], and Face $(316/40)$, Shark $(240/91)$ and Walking $(260/55)$ in [25], where (F/P) denotes the number of frames (F) and points (P).

[1] An identical rearrangement was used in [3] but with different motivation and for different purpose.

[2] We did not use the CSF variant of [15] as they are very similar.

5.2. Cumulative histograms of errors

Our first experiment is aimed to give a statistical comparison between the performance of our method and several existing methods. For this purpose we use a real motion capture sequence, here *e.g.* the Stretch sequence. From the ground-truth 3D point clouds of the sequence as well the true camera matrices, we re-synthesize F frames of image measurements with Gaussian random noise added in, where noise ratio is defined as $\|\text{Noise}\|_{\text{Fro}}/\|\mathtt{W}\|_{\text{Fro}}$. Use the obtained data, we test our methods, as well as several other existing methods. We repeat the random test 100 times. Then, we plot the cumulative histograms of the rotation estimation errors, and the 3D reconstruction errors, as shown in Fig. 1. This figure clearly reveals that: our block-matrix method outperforms most of the other methods, and our pseudo-inverse method also achieves better results compared with EM-PPCA and XCK.



(a) (b)

Figure 1. Cumulative histograms of errors tested on the "Stretch" sequence. The most left curve gives the best performance. **Left**: the rotation error; **Right**: the 3D reconstruction error. Our Block-Matrix method outperforms most of the other existing methods.

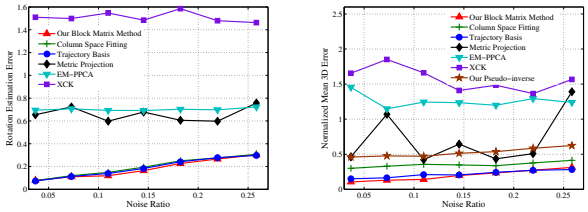
5.3. Noise performance

To analyze the behavior of our new methods under noise, we repeat the (above) first experiment at different noise ratios. Example results on the Stretch sequence are given in Fig. 2 which plots the estimation errors as a function of the noise ratio.

It is seen, our block matrix method achieves the best performance in terms of the accuracy for rotation estimation and for shape recovery, compared favorably with almost all the other state-of-the-art competitors. Note that our pseudo-inverse method also achieves better performance than EM-PPCA, Metric Projection and XCK.

5.4. Compare all methods on all real sequences

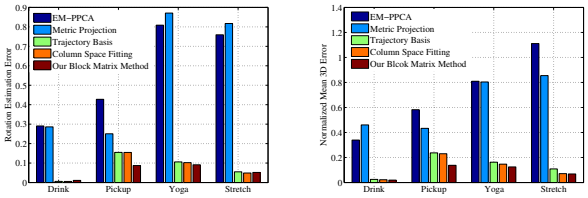
In this subsection, we provide experimental results of all the 5 methods we are benchmarking, on all the real sequences at hand. Table-1 summarizes our main results, where both the shape reconstruction error (mean 3D error) and the camera rotation error are provided (whenever the ground truth are available). Fig.-3 shows the comparison.



(a) (b)

Figure 2. Noise performance. (a) Rotation estimation error. (b) Normalized mean 3D error.

Clearly, our block matrix method achieves the best performance in shape recovery, on almost all of the benchmark sequences (the Shark sequence is an exception, but that possibly due to the fact that this Shark sequence is in fact degenerate [25]).



(a) (b)

Figure 3. Motion capture data experimental results. **Left**: Rotation estimation error; **Right**: 3D reconstruction error.

5.5. Test on frame-reshuffled data

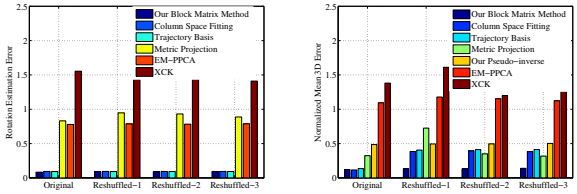
A highlight of this work is that our method does not assume any prior knowledge about the problem. For instance, we do not assume the trajectories are smooth across frames, while many other method do make this assumption, either explicitly or implicitly. So, we expect (predict) that our method is immune to random frame-order reshuffle (permutation).

To verify this point, we redo the experiments but on frame-reshuffled data, and obtain the following results in Fig.-4. It is seen that both the trajectory basis method and the CSF method perform very badly on the reshuffled sequence, while our method remains unaffected.

Fig.-5 gives a close inspection. The top row compares the trajectories recovered by using the original sequence, and using a frame-reshuffled sequence, while the bottom row shows the normalized mean 3D error for both sequences. From this figure, the trajectory basis method fails to output acceptable results on the frame-permutated sequence, but our method leads to identical results on both cases. This is not surprising as permutating a matrix will not change its rank, and our methods does not assume any frame order or temporal smoothness.

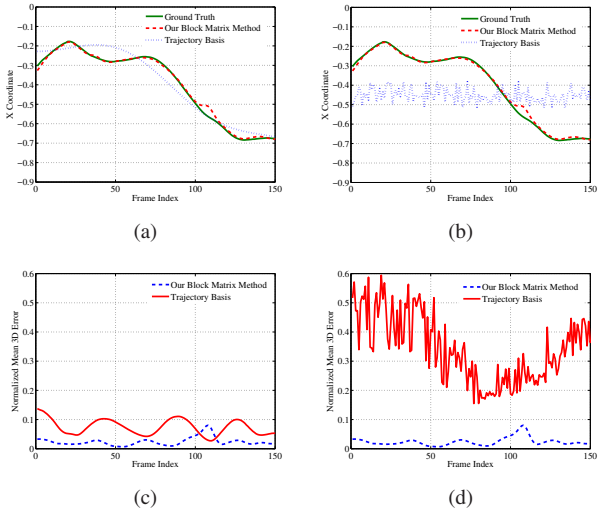
Table 1. Quantitative comparison of our proposed methods versus the state-of-the-art methods on benchmark video sequences. e_{3D}(P) and e_{3D}(B) denote the 3D errors of our Pseudo-inverse method and Block matrix method respectively.

Dataset	XCK		EM-PPCA		Metric Projection		Trajectory Basis		Column Space Fitting		Proposed Methods		
	e_R	e_{3D}	e_R	e_{3D}	e_R	e_{3D}	e_R	e_{3D}(K)	e_R	e_{3D}(K)	e_R	e_{3D}(P)	e_{3D}(B)
Drink	0.336	3.519	0.291	0.339	0.286	0.460	0.006	0.025(13)	0.006	0.022(6)	0.011	0.451(4)	**0.019(4)**
Pick-up	0.469	3.372	0.428	0.582	0.251	0.433	0.155	0.237(12)	0.155	0.230(6)	0.087	0.580(7)	**0.138(7)**
Yoga	1.201	7.494	0.809	0.810	0.871	0.804	0.106	0.162(11)	0.102	0.147(7)	0.091	0.659(9)	**0.125(9)**
Stretch	0.949	4.242	0.759	1.111	0.817	0.855	0.055	0.109(12)	0.049	0.071(8)	0.052	0.468(8)	**0.069(8)**
Dance	-	2.996	-	0.984	-	0.264	-	0.296(5)	-	0.271(2)	-	0.575(10)	**0.171(10)**
Face	-	-	-	0.033	-	0.036	-	0.044(5)	-	0.036(3)	-	0.485(7)	**0.030(7)**
Walking	-	-	-	0.492	-	0.561	-	0.395(2)	-	0.186(2)	-	0.471(6)	**0.132(6)**
Shark	-	-	-	0.050	-	0.157	-	0.180(9)	-	0.008(3)	-	0.902(3)	0.242(3)



(a) (b)

Figure 4. Performance on frame order reshuffled sequences. (a) Rotation error. (b) Normalized mean 3D error. (**Better viewed in color**).



(a) (b)

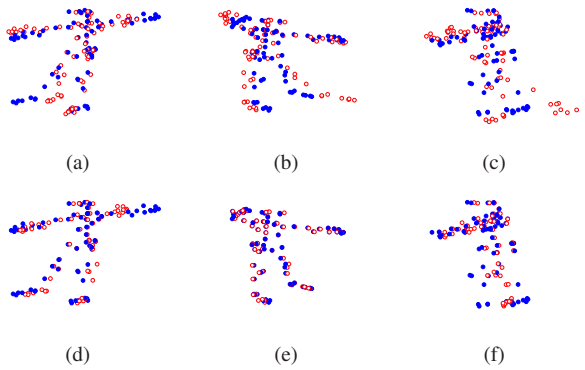
(c) (d)

Figure 5. Comparison of our block matrix method versus the trajectory basis method, on an original input video sequence, as well as on a frame-reshuffled version. (a) Recovered trajectory of one point in the X coordinate by both methods on the original sequence. (b) Recovered trajectory of one point in the X coordinate by both methods on the frame-reshuffled sequence. (c) Normalized mean 3D error for both methods on the original sequence. (d) Normalized mean 3D error for both methods on the frame-reshuffled sequence.

5.6. Sample shape reconstruction results

For visual evaluation, we give result comparison between our block matrix method and the trajectory basis method on a more complex sequence, Dance, see Fig.-6.

We also test the Talking Face video[3], using 500 frames and 68 feature tracks. Fig.-7 shows 3 frames of the original images and the resulted 3D points, where the reprojection error is 0.9222 pixels.



(a) (b) (c)

(d) (e) (f)

Figure 6. Comparison of the 3D reconstruction results on the Dance sequence. The blue dots are the ground truth 3D points, and the red circles show the reconstructed points. **Top row:** results by the trajectory basis method [3], where the 3D errors are **0.3011, 0.2827, 0.2814** for the 3 frames. **Bottom row:** our result by the block matrix method, where the 3D errors are **0.2228, 0.0355, 0.1389** for the 3 frames.

6. Closing remarks

This paper advocates a novel prior-free approach to non-rigid factorization. Our method is purely convex, very easy to implement, and is guaranteed to converge to an optimal solution (at least approximately up to certain relaxation). It shows that, contrary to common belief, the NRSFM factorization problem can be solved unambiguously, efficiently and accurately, without using extra priors. This said, however, from a practical point of view, we do not against the use of available prior, as long as the prior is sensible and

[3]http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/

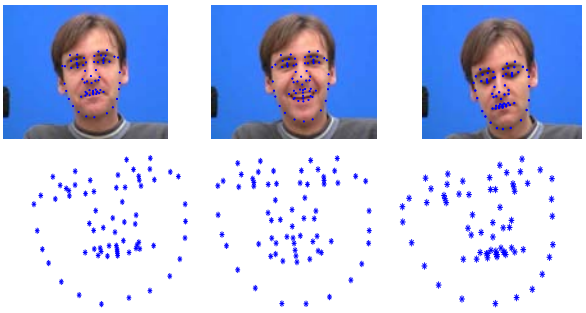


Figure 7. Example 3D deformable shape reconstruction results. **Top row**: sample frames of the input Face sequence. **Bottom row**: recovered 3D face shapes by using our block matrix method.

reflects the physical nature of the problem at hand. It is expected that, using good prior will further improve our solution, and make our method more applicable to complex scenarios.

In the present paper, we have concentrated on complete measurement case under orthographic camera model. Thanks to recent progress in SFM and Compressive Sensing, the proposed method can be easily adapted to handling missing-data case (*e.g.* [8, 10, 12]), outlier case (*e.g.* [9, 18, 4]), multibody motion case [17], as well as perspective camera case [10].

Acknowledgement. This work is funded, in part by Natural Science Foundation of China (60736007), and by Australia Research Council. The authors wish to thank Richard (Prof. Hartley) for invaluable discussions.

References

[1] H. Aanæs and F. Kahl. Estimation of deformable structure and motion. In *Workshop on Vision and Modelling of Dynamic Scenes, ECCV*, pages 1–4, 2002. 2

[2] I. Akhter, Y. Sheikh, and S. Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *CVPR*, pages 1534–1541, 2009. 2, 4

[3] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*, 2008. 1, 2, 4, 5, 7

[4] R. Angst, C. Zach, and M. Pollefeys. The generalized trace-norm and its application to structure-from-motion problems. In *ICCV*, pages 1–8, 2011. 3, 8

[5] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *CVPR*, pages 1–8, 2008. 1, 2

[6] M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *CVPR*, 2005. 2, 4

[7] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, pages 690–696, 2000. 1, 2

[8] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *CVPR*, pages 316–322, 2005. 8

[9] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *J. ACM*, 58(3):11:1–37, 2011. 8

[10] Y. Dai, H. Li, and M. He. Element-wise factorization for n-view projective reconstruction. In *ECCV*, pages 396–409, 2010. 8

[11] A. Del Bue. A factorization approach to structure from motion with shape priors. In *CVPR*, pages 1–8, 2008. 1

[12] A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L_1 norm. In *CVPR*, pages 771–778, 2010. 8

[13] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In *ECCV*, pages 297–310, 2010. 2

[14] P. Gotardo and A. Martinez. Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. *PAMI*, 33(10):2051–2065, 2011. 1, 2, 4, 5

[15] P. Gotardo and A. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR*, pages 3065–3072, 2011. 5

[16] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *ECCV*, pages 276–289, 2008. 2

[17] H. Li. Two-view motion segmentation from linear programming relaxation. In *CVPR*, pages 1 –8, june 2007. 8

[18] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *ICCV*, pages 1074–1080. IEEE, 2009. 8

[19] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *CoRR*, abs/1010.2955, 2010. 4

[20] S. Ma, D. Goldfarb, and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming, Series A*, 128(1,2):321–353, 2011. 5

[21] M. Paladini, A. D. Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. In *CVPR*, pages 2898–2905, 2009. 1, 4, 5

[22] V. Rabaud and S. Belongie. Linear embeddings in non-rigid structure from motion. In *CVPR*, pages 2427–2434, 2009. 2

[23] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010. 3

[24] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, pages 2761–2768, 2010. 2

[25] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30, 2008. 1, 4, 5, 6

[26] J. Xiao, J.-x. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *ECCV*, pages 573–587, 2004. 1, 2, 3, 4, 5