# Sequential Factorization for Nonrigid Structure from Motion via LBFGS

Qiulei Dong[1,3] and Hao Hu[2,1]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Information Science and Engineering, Shan Dong University, Jinan 250100, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
qldong@nlpr.ia.ac.cn, huhaosdu@163.com

***Abstract*—How to implement an effective factorization for nonrigid structure from motion(NRSFM) has attracted much attention in recent years. Addressing this problem, we propose a novel sequential factorization method without extra priors other than the basis low-rank prior, consisting of a motion estimation module and a 3D shape recovery module. In the motion estimation module, for improving the estimation accuracy, a novel objective function is designed for jointly pursuing the Euclidean corrective matrix and the shape coefficient matrix. And an iterative minimization algorithm is explored to solve the designed objective function based on the Limited-memory Broyden-Fletcher-Goldfarb-Shanno approach(LBFGS), naturally leading to the rotation matrix. In the 3D shape recovery module, a simple iterative algorithm is introduced for effectively computing the 3D deformable shapes with the estimated rotation matrix. The proposed extra-prior-free method is easy to implement and it can achieve an effective tradeoff between estimation accuracy and computational speed, since only several classic techniques are involved. Extensive experimental results demonstrate the effectiveness of the proposed method in comparison to five state-of-the-art methods.**

## I. Introduction

Nonrigid structure from motion(NRSFM) has attracted much attention in recent years, which is to recover unknown camera motion matrices and 3D deformable shapes of an object from 2D point tracks. NRSFM is a typically under-constrained problem, since the number of the observed elements are far less than the number of unknown variables. To overcome it, a lot of works with different assumptions on 3D deformable shapes have been explored to cast the original problem to an over-constrained optimization problem[1], [2], [3], [4], [5].

A widely-used assumption for handling the NRSFM problem is the low-rank prior, which assumes that 3D deformable shapes of an object can be modelled in a low-rank subspace. Let $W \in \mathcal{R}^{2f\times p}$($W$ has been centralized) represent an input matrix where $p$ 2D point tracks from $f$ frames are stacked. Under the orthographic camera model, a general NRSFM factorization method aims to factorize $W$ into a shape matrix $S \in \mathcal{R}^{3f\times p}$ and a block-diagonal camera motion matrix $R = \mathrm{diag}(R_1, ..., R_i, ..., R_f) \in \mathcal{R}^{2f\times 3f}$, where $R_i \in \mathcal{R}^{2\times 3}(i = 1, 2, ..., f)$ represents a 3D rotation followed by an orthographic projection. Bregler et al.[1] firstly proposed a factorization method with the low-rank prior under the orthographic camera model, assuming that the shape matrix $S$ can be represented by a combination of $k$ basis shapes in a low-rank shape subspace. Let $B \in \mathcal{R}^{3k\times p}$ represent the shape basis matrix consisting of $k$ basis shapes, and $C \in \mathcal{R}^{f\times k}$ the coefficient matrix. Then, the method [1] factorizes $W$ into a factor matrix $L$ and a shape basis matrix $B$ as:

$$W = RS = R(C \otimes I_3)B = LB \tag{1}$$

where '$\otimes$' indicates the Kronecker product, $I_3$ is the $3 \times 3$ identity matrix, and $L = R(C \otimes I_3) \in \mathcal{R}^{2f\times 3k}$. Obviously, it holds that $\mathrm{rank}(S) \leq 3k$. Then following this work[1], many NRSFM methods have been derived from the formulation (1).

By utilizing the low-rank prior on the shape subspace (i.e. the low-rank shape representation) in [1], many methods have been proposed for improving the factorization accuracy for NRSFM further. The alternating multi-linear optimization framework[6], [7], [8], [9], [10] for NRSFM has been continually investigated, where each of these unknown variables is estimated by fixing the others at each iterative step. Dai et al.[11], [12] proposed an effective extra-prior-free NRSFM method (called BMM), which achieved comparable performances with some existing extra-prior-based methods. In addition, Akhter et al.[13] investigated the dual form of the low-rank shape representation, and used a set of Discrete Cosine Transform(DCT) bases to recover the nonrigid structure of an object, which implicitly relied on the smoothness prior. Gotardo and Martinez[14] proposed a 3D shape trajectory method in the DCT domain, assuming the object structure to smoothly deform over time. Hamsici et al.[15] presented a kernel-based NRSFM method with a spatial-smoothness prior, which can deal with frame-reshuffled data effectively.

Different from the existing methods derived from (1), Lee et al. [16] proposed an EM-based NRSFM method under a defined Procrustean normal distribution model. Cho et al. [17] proposed a Procrustean normal distribution mixture model for handling NRSFM problems.

It is noted that it is hard for many existing methods to achieve an effective tradeoff between estimation accuracy and computational speed when large-scale data is handled(e.g. [16], [11], [12], [14], [15] as shown in Section III). Addressing this problem, under the orthographic camera model, we propose a novel NRSFM method without extra priors, which calculates the Euclidean corrective matrix(defined in Section II), the camera motion matrix, and the shape matrix sequentially. In this method, the Euclidean corrective matrix in an explicit form is iteratively computed by an explored minimization algorithm, and then the shape matrix is computed by solving a rank-constraint-based optimization problem via an explored modified version of the SVP(Singular Value Projection)-Newton algorithm[18].

The main advantages of the proposed method are: (i) It is quite easy to implement, since only several classic techniques are involved. (ii) Compared with five state-of-the-art methods, it performs better on both noise-free and noisy data in most cases, and it can achieve an effective tradeoff between estimation accuracy and computational speed, as demonstrated by our experimental results in Section III.

## II. Sequential Factorization Method without Extra Priors

Let $S_i = [s_{i,1}, s_{i,2}, ..., s_{i,p}] \in \mathcal{R}^{3\times p}$ $(i = 1, 2, ..., f)$ be the 3D points in the $i$-th frame, where $s_{i,j} = [x_{i,j}, y_{i,j}, z_{i,j}]^T (j = 1, 2, ..., p)$ is the $j$-th 3D point in the $i$-th frame. Let $S^b \in \mathcal{R}^{f\times 3p}$ represent a rearrangement of the shape matrix $S$, the $i$-th row of which has the form $[x_{i,1}, x_{i,2}, ..., x_{i,p}, y_{i,1}, y_{i,2}, ..., y_{i,p}, z_{i,1}, z_{i,2}, ..., z_{i,p}]$, i.e. $S^b = [I_X, I_Y, I_Z](I_3 \otimes S)$ where $I_X$,$I_Y$,$I_Z$ respectively represent the submatrices consisting of all the $(3i-2)$th,$(3i-1)$th, $(3i)$th rows of the $3f$-order identity matrix $I_{3f}$. Since $S = (C \otimes I_3)B$ under the rank-$k$ model (1), we have $S^b = C[I_x, I_y, I_z](I_3 \otimes B)$ where $I_x$,$I_y$,$I_z$ respectively represent the submatrices consisting of all the $(3j-2)$th, $(3j-1)$th, $(3j)$th $(j = 1, 2, ..., k)$ rows of the $3k$-order identity matrix $I_{3k}$. Obviously, it holds that $\mathrm{rank}(S^b) \leq \mathrm{rank}(C) \leq k$ [11], which reflects the essence of the rank-$k$ model compared with the condition $\mathrm{rank}(S) \leq 3k$. Then, under the orthographic camera model, the camera motion matrix $R$ and the shape matrix $S$ can be estimated by solving the following unified low-rank minimization problem(or its variants) with a specified rank $k$:

$$\arg\min_{R,S} ||W - RS||_F^2, s.t.\ S \in \mathcal{K} = \{S|\mathrm{rank}(S^b) \leq k\} \quad (2)$$

It is noted that (2) is non-convex, so it is hard to obtain a global minimum solution to it. Here, aiming to pursue an effective local minimum solution, we propose a sequential extra-prior-free factorization method under the basic low-rank shape model (compatible with the dual trajectory model), where the Limited-memory Broyden-Fletcher-Goldfarb-Shanno(LBFGS) approach [19] is employed, named as SFLBFGS.

As seen from (1), the input matrix $W$ can be approximately factorized into the product of $\hat{L} \in \mathcal{R}^{2f\times 3k}$ and $\hat{B} \in \mathcal{R}^{3k\times p}$ via truncated SVD where the result of SVD on $W$ is truncated to the largest $3k$ singular values, i.e. $W \approx \hat{L}\hat{B}$. Obviously, this decomposition is determined up to a nonsingular $3k \times 3k$ matrix. Let $G \in \mathcal{R}^{3k\times 3k}$ be a nonsingular matrix that upgrades $\hat{B}$ into a canonical Euclidean shape basis matrix $B$, called the Euclidean corrective matrix, then according to (1), we have

$$R(C \otimes I_3) = L = \hat{L}G, \quad B = G^{-1}\hat{B} \quad (3)$$

The SFLBFGS method consists of two modules, the motion estimation module and the 3D shape recovery module. Firstly, the motion estimation module is implemented for jointly estimating the Euclidean corrective matrix $G$ and the coefficient matrix $C$ by solving a minimization problem derived from (3) (In fact, we only calculate sub-blocks $G_m, c_m$ of $G, C$ as explained in Section II-A), then for computing the camera motion matrix $R$. Secondly, the 3D shape recovery module is implemented for computing the shape matrix $S$ by an explored modified version of the SVP-Newton algorithm [18].

### A. Motion Estimation Module

For a sequential method, it is important to calculate an accurate Euclidean corrective matrix $G$ since the rest variables are all computed based on it. Let $\hat{L}_{2i-1:2i} (i = 1, 2, ..., f)$ denote the $(2i-1)$-th and $2i$-th rows of $\hat{L}$. Let $G_m$ and $c_m$ denote the $m$-th column-triplet of $G$ and the $m$-th column of $C$ respectively. Since $R_i R_i^t = I_2$, the following constraint on $G_m$ is obtained from (3) as:

$$\hat{L}_{2i-1:2i} G_m G_m^T \hat{L}_{2i-1:2i}^T = c_{im} R_i R_i^T c_{im} = c_{im}^2 I_2 \quad (4)$$

where $I_2$ is the $2 \times 2$ identity matrix, and $c_m = [c_{1m}, c_{2m}, ..., c_{fm}]^T$ is a not-all-zero vector.

Many existing sequential methods constructed different objective functions for estimating $G_m$ according to (4). In [13], [14], [15], $G_m$ was directly estimated according to (4) with a set of preseted coefficients $c_m$ via nonlinear optimization. However, it is hard to manually preset a set of appropriate coefficients $c_m$ for estimating $G_m$. Dai et al. [11], [12] eliminated $c_m$ from (4) and used the Gram matrix $Q_m = G_m G_m^T$ as a variable matrix replacing $G_m$. Accordingly, they constructed a trace-minimization problem on $Q_m$, and solved it via SDP (Semi-Definite Programming). However, the obtained $Q_m$ by SDP is usually not a rank-3 matrix so that $G_m$ has to be computed via truncated SVD, which means that the estimated $G_m$ is just an approximate and inaccurate solution to (4). Then, a non-linear refinement on $G_m$ was employed in [11], [12] to improve the estimation accuracy on $G_m$. It has to be pointed out that although the size of the involved SDP is not quite big, the computational cost is still relatively high, especially when the specified rank $k$ is increased.

Addressing these above problems, we do not eliminate(or preset) $c_m$ here, and we also do not replace $G_m G_m^T$ with $Q_m$, but construct the following minimization problem for jointly estimating $G_m$ and $c_m$ in explicit forms according to (4). For computational convenience, an auxiliary vector $b_m = [b_{1m}, ..., b_{im}, ..., b_{fm}]^T$ is introduced where $b_{im} = c_{im}^2 \geq 0 (i = 1, 2, ..., f)$:

$$\min_{G_m, b_m} \quad \mathcal{F} = \sum_{i=1}^{f} ||\hat{L}_{2i-1:2i} G_m G_m^T \hat{L}_{2i-1:2i}^T - b_{im} I_2||_F^2 \quad (5)$$
$$s.t. \quad ||b_m||_F^2 = f, \quad b_m \geq 0$$

where the constraint $||b_m||_F^2 = f$ is used to fix the scale freedom and exclude the all-zero solution.

The objective function (5) has two important characteristics: (i) Compared with the existing Gram-matrix-based methods, it avoids the positive semidefinite constraint which is expensive

**Algorithm 1**: Algorithm for estimating $G_m$ and $b_m$

```
Input: $b_{m(0)} = \mathbf{1}$($\mathbf{1}$ indicates an all-one vector), $t = 0$
Output: $G_m$, $b_m$
while not converge do
    Fix $b_{m(t)}$ and update $G_{m(t+1)}$ via LBFGS ;
    Fix $G_{m(t+1)}$ and update $b_{m(t+1)}$ according to (9);
    $t = t + 1$ ;
end
```

to handle, as well as the gap between the original rank-$k$ optimization problem and other relaxed variants (e.g. the used nuclear-norm relaxation in [11]). (ii) Compared with the existing methods (e.g. [13], [14], [15]) estimating $G_m$ directly via nonlinear optimization with a set of preseted coefficients $c_m$, it is able to adaptively determine these coefficients for further improving shape recovery accuracy and robustness.

Here, an iterative algorithm is proposed for estimating $G_m$ and $b_m$. At each iteration, each of the two variables is updated by fixing the other one. The detailed performance of the iterative procedure is described as follows, and the complete algorithm is outlined in Algorithm 1.

**Update** $G_m$**:** Fixing $b_{m(t)}$(the subscript '$t$' indicates the time index), we have the following problem from (5):

$$\begin{aligned} G_{m(t+1)} &= \arg\min_{G_m} \mathcal{F} \\ &= \arg\min_{G_m} \sum_{i=1}^{f} ||\hat{L}_{2i-1:2i} G_m G_m^T \hat{L}_{2i-1:2i}^T - b_{im(t)} I_2||_F^2 \end{aligned} \tag{6}$$

For solving (6), we employ the Limited-memory Broyden-Fletcher-Goldfarb-Shanno(LBFGS) approach [19] with a strong Wolfe-Powell line search, which is quite effective for optimizing such a type of objective function as demonstrated in [20]. The LBFGS approach requires the gradient $\nabla_{G_m}\mathcal{F}$ of $\mathcal{F}$ with respect to $G_m$ as

$$\begin{aligned} &\nabla_{G_m}\mathcal{F} = \\ &4\sum_{i=1}^{f}(\hat{L}_{2i-1:2i}^T\hat{L}_{2i-1:2i}G_mG_m^T - b_{im(t)}I_{3k})\hat{L}_{2i-1:2i}^T\hat{L}_{2i-1:2i}G_m \end{aligned} \tag{7}$$

**Update** $b_m$**:** Fixing $G_{m(t+1)}$, let $Z_1$ be the column vector whose $i$-th element is $\hat{L}_{2i-1}G_{m(t+1)}G_{m(t+1)}^T\hat{L}_{2i-1}^T$, and let $Z_2$ be the column vector whose $i$-th element is $\hat{L}_{2i}G_{m(t+1)}G_{m(t+1)}^T\hat{L}_{2i}^T$. Obviously, $Z_1$ and $Z_2$ are non-negative, and then we have the following problem from (5):

$$\begin{aligned} &b_{m(t+1)} = \arg\min_{b_m}\mathcal{F} = \arg\min_{b_m} ||Z_1 - b_m||_F^2 + ||Z_2 - b_m||_F^2 \\ &s.t. \quad ||b_m||_F^2 = f, \quad b_m \geq 0 \end{aligned} \tag{8}$$

Since both $Z_1$ and $Z_2$ are non-negative, there exists the following closed-form solution to (8):

$$b_{m(t+1)} = \sqrt{f}(Z_1 + Z_2)/||Z_1 + Z_2||_F \tag{9}$$

**Calculate** $R$**:** Once $G_m$ and $b_m$ are obtained, the motion matrix $R$ can be directly estimated according to $\hat{L}_{2i-1:2i}G_m = c_{im}R_i$ and $b_{im} = c_{im}^2$, and its sign ambiguity is handled as done in [11], [21].

**Algorithm 2**: Modified version of SVP-Newton

```
Input: $W$, $R$, $\mu$, $S_{(0)} = \mathbf{0}$($\mathbf{0}$ indicates an all-zero matrix), $t = 0$
Output: $S$
while not converge do
    $Y_{(t+1)} = S_{(t)} - \mu\nabla\Phi(S_{(t)})$ ;
    $(U_{(t+1)}, V_{(t+1)}) \leftarrow \Pi_{\mathcal{K}}(Y_{(t+1)})$ via SVD on $Y_{(t+1)}^b$ ;
    $\Sigma_{(t+1)}$ by minimizing (13) ;
    $S_{(t+1)} \leftarrow S_{(t+1)}^b = U_{(t+1)}\Sigma_{(t+1)}V_{(t+1)}^T$ ;
    $t = t + 1$ ;
end
```

**Remarks:** SFLBFGS does not compute $G_m$ by solving (5) with every $m(m = 1, 2, ..., k)$, but only computes $G_m$ with a specified $m$(without loss of generality, denote it as $G_1$). As demonstrated in [11], [13], [14], [15], it is feasible to compute the camera motion matrix only with the column-triplet $G_1$ instead of the whole Euclidean corrective matrix $G$.

### B. 3D Shape Recovery Module

With the obtained $R$ above, SFLBFGS calculates $S$ with a specified rank $k$ by solving the following rank-constraint-based problem on $S$ without additional regularizers according to (2) as:

$$\begin{aligned} &\min_S \Phi = ||W - RS||_F^2 \\ &s.t. \ S \ \in \mathcal{K} = \{S \mid \text{rank}(S^b) \leq k\} \end{aligned} \tag{10}$$

We propose a modified version of the SVP-Newton algorithm [18] for solving (10). At each iteration, a gradient descent update is firstly implemented as:

$$Y_{(t+1)} = S_{(t)} - \mu\nabla\Phi(S_{(t)}) \tag{11}$$

where $\nabla\Phi(S_{(t)}) = 2R^T(RS_{(t)} - W)$ is the gradient of $\Phi$ with respect to $S$ at time $t$, and $\mu$ is the step size. Then $Y_{(t+1)}$ is projected to the constraint set $\mathcal{K}$ by the following projection operator $\Pi_{\mathcal{K}}(\cdot)$ as:

$$\begin{aligned} &\Pi_{\mathcal{K}}(Y_{(t+1)}) = \arg\min_S \quad ||S - Y_{(t+1)}||_F^2 \\ &s.t. \quad S \in \mathcal{K} = \{S \mid \text{rank}(S^b) \leq k\} \end{aligned} \tag{12}$$

By rearranging $Y_{(t+1)}$ into $Y_{(t+1)}^b$ in (12) under the same way as rearranging $S_{(t)}$ into $S_{(t)}^b$ and implementing SVD on $Y_{(t+1)}^b$, $S_{(t+1)}^b$ can be calculated according to the top $k$ singular values $\Sigma_{(t+1)}$ and the corresponding vectors $(U_{(t+1)}, V_{(t+1)})$ of $Y_{(t+1)}^b$.

In order to further speed up this algorithm's convergence, a Newton-type step is introduced to update $\Sigma$ according to the obtained $(U_{(t+1)}, V_{(t+1)})$ by solving the following minimization problem:

$$\min_{\Sigma}\Phi = ||W - RS_{(t)}||_F^2, \ s.t. \ S_{(t)}^b = U_{(t+1)}\Sigma V_{(t+1)}^T \tag{13}$$

Via a set of mathematical transformations, (13) can be transformed into a standard least-squares problem, and a closed-form solution can be obtained further. Due to limited space, the details for updating $\Sigma$ are provided in the supplementary file, and the complete algorithm for calculating $S$ is outlined in Algorithm 2. In addition, similar to the SVP-NewtonD[18], a natural constraint restricting $\Sigma$ to be diagonal can also be introduced to (13) for reducing computational costs further.

### C. Algorithmic Analysis

Algorithm 1 is designed under a standard iterative optimization framework involving a least-squares step and a limited memory BFGS step, hence, its convergence is dependent on the convergence of the limited memory BFGS algorithm for solving (6) with respect to $G_m$. Nocedal and Wright [22] have given a strict proof for the superlinear convergence of BFGS when it applies to general nonlinear (not just convex) objective functions. Moreover, it is noted from Algorithm 1, the main computational cost of Algorithm 1 is spent on updating $G_m$ via LBFGS that has a quasi-Newton update speed and simultaneously requires only $O(k^2)$ memory [19]. Moreover, it is noted from (6) that the running time for updating $G_m$ is strongly related to both the selected rank $k$ and the frame number $f$. With the increase of $k$ and $f$, the running time of Algorithm 1 is increased accordingly.

The difference between Algorithm 2 and the standard SVP-Newton algorithm is that: the unknown variable of the objective function (10) is $S$ and the unknown variable of the corresponding rank constraint is $S^b$ in Algorithm 2, while the unknown variable of both the objective function and the corresponding rank constraint is the same $S$ in the standard SVP-Newton algorithm. Accordingly, since $S^b$ has a different form from $S$ (up to a transformation, i.e. $S^b = [I_X, I_Y, I_Z](I_3 \otimes S)$), an additional set of mathematical transformations has to be implemented for updating the singular value matrix  by minimizing (13) in Algorithm 2 in contrast to the standard SVP-Newton algorithm. However, Algorithm 2 is indeed a straightforward variant of the SVP-Newton algorithm whose convergence is proved in [18]. Similar convergent guarantee still holds for Algorithm 2 via a straightforward modification of the corresponding proof in [18].

## III. Experimental Results

We test the proposed SFLBFGS as well as five state-of-the-arts: PTA[13], CSF[14], RIKA[15], BMM[12], EM-PND[16], where PTA, CSF, and RIKA assume additional smoothness priors besides the low-rank prior. Experimental results are evaluated by using the same error metrics as reported in [11], [12], [14]: Rotation estimation error $e_R = \frac{1}{f}\sum_{i=1}^{f}||R_i - \hat{R}_i||_F$ is to measure the average errors between the true rotations $R_i$ and aligned estimated rotations $\hat{R}_i$; 3D reconstruction error $e_{3D} = \frac{1}{\sigma_{3D} f p}\sum_{i=1}^{f}\sum_{j=1}^{p} e_{ij}$ ($\sigma_{3D} = \frac{1}{3f}\sum_{i=1}^{f}(\sigma_{ix} + \sigma_{iy} + \sigma_{iz})$) is to measure average normalized 3D errors between the true 3D points and aligned reconstructed 3D points, where $e_{ij}$ indicates the Euclidean reconstruction error for the $j$-th point at the $i$-th frame, $\sigma_{ix}$, $\sigma_{iy}$, $\sigma_{iz}$ denote the standard deviations respectively of the $X$, $Y$, and $Z$ coordinates of the true 3D shape at the $i$-th frame.

Initials for SFLBFGS are: $G_{k(0)} = [1,0,0;\cdots;0,1,0;\cdots;0,0,1]$, $b_{(0)} = \mathbf{1}$, $S_{(0)} = \mathbf{0}$. As done in [11], [12], [13], [14], [15], we implement the referred methods with different ranks (Except EM-PND[16] that is not dependent on a manually specified rank) and report their best results.

### A. Performances on Benchmark Data

This subsection evaluates the performances of the referred methods on the following sequences that are extensively used for evaluating NRSFM methods, where the true 3D shapes (also the true rotation matrices in the first four sequences) are given: Drink(1102/41), Pickup(357/41), Yoga(307/41), Stretch(370/41), Dance(264/75)[13]; Face1(74/37)[23]; Face2 (316/40), Shark(240/91), Walking(260/55)[2], Capoeira (250/ 41)[24], where ($f/p$) after the sequence name indicates the number of frames and 3D points. Moreover, the Flag(60/4800) sequence[25] and the UMPM(5168/15) sequence[26] are also used to evaluate the performances of the referred methods on more complex large-scale data.

Table I reports the results of the referred methods on these sequences. Here, two points about EM-PND [16] need to be explained: (i) Although EM-PND performs well on several sequences, it cannot effectively handle 2D data corresponding to a set of different 3D rotation matrices and deformable shapes, such as Drink, Pickup, Yoga, Stretch. This is not conflict with the experimental results in [16], since the used ground truth shapes for these four sequences in the code package from the authors [16] are obtained by rotating the original ground truth shapes (from [13]) with the ground truth rotation matrices in advance. That is to say, in the four processed sequences by EM-PND, there exists no 3D rotation but only an orthographic projection between the input 2D data and the corresponding 3D shapes. To make available rotation comparisons, the original forms of all the sequences are used as done in [11], [12], [13], [14], [15]. The symbol '□' in Table I(also Table IV) indicates this case. (ii) Under the current hardware configuration of the used PC, EM-PND cannot deal with the Flag sequence that needs an oversize computational memory for EM-PND, and the symbol '◇' in Table I(also Table IV) indicates this case.

In order to further analyze the effectiveness of the two modules in SFLBFGS, we test these methods again on the Drink, Pick-up, Yoga, Stretch sequences where the real rotation matrices are known. The lowest rotation estimation errors regardless of 3D shape recovery accuracy by these methods are listed in Table II (their errors and best ranks may not be equal to those corresponding to the best reconstruction errors in Table I). Moreover, these methods are implemented for estimating the 3D shapes with the real rotation matrices instead of their estimated rotation matrices, and their corresponding 3D reconstruction errors are reported in Table III.

As seen from Tables I–III, the designed two modules and the whole SFLBFGS method achieve better performances than the rest methods in most cases, which demonstrates the effectiveness of SFLBFGS for handling the benchmark data.

In addition, in order to investigate the relationship between the data size and the computational cost of each method more conveniently, the running times of the referred methods on the

TABLE I

COMPARISON ON BENCHMARK SEQUENCES ($K$ INDICATES THE SELECTED BEST RANK).

| Methods | PTA | | CSF | | RIKA | | BMM | | EM-PND | | SFLBFGS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $e_R$ | $e_{3D}$ | $e_R$ | $e_{3D}$ | $e_R$ | $e_{3D}(K)$ | $e_R$ | $e_{3D}(K)$ | $e_R$ | $e_{3D}$ | $e_R$ | $e_{3D}(K)$ |
| Drink | 0.006 | 0.025(13) | 0.006 | 0.022(6) | 0.006 | 0.027(6) | 0.007 | 0.027(12) | □ | □ | 0.006 | **0.017(13)** |
| Pick-up | 0.155 | 0.237(12) | 0.155 | 0.230(6) | 0.155 | 0.231(3) | 0.121 | 0.173(12) | □ | □ | 0.105 | **0.168(13)** |
| Yoga | 0.106 | 0.162(11) | 0.102 | 0.147(7) | 0.102 | 0.152(7) | 0.088 | 0.115(10) | □ | □ | 0.085 | **0.111(13)** |
| Stretch | 0.055 | 0.109(12) | 0.049 | 0.071(8) | 0.049 | 0.086(8) | 0.068 | 0.103(11) | □ | □ | 0.049 | **0.068(13)** |
| Dance | – | 0.296(5) | – | 0.271(2) | – | 0.173(7) | – | 0.186(10) | – | 0.241 | – | **0.156(25)** |
| Face1 | – | 0.125(3) | – | 0.064(5) | – | 0.069(5) | – | 0.050(11) | – | **0.036** | – | **0.036(10)** |
| Face2 | – | 0.044(5) | – | 0.036(3) | – | 0.032(4) | – | 0.030(7) | – | **0.023** | – | 0.026(13) |
| Walking | – | 0.395(2) | – | 0.186(2) | – | 0.104(5) | – | 0.130(8) | – | **0.069** | – | 0.095(12) |
| Shark | – | 0.180(9) | – | **0.008(3)** | – | 0.101(3) | – | 0.231(4) | – | 0.024 | – | 0.176(2) |
| Capoeira | – | 0.507(6) | – | 0.341(4) | – | 0.439(4) | – | 0.393(5) | – | 0.514 | – | **0.245(13)** |
| Flag | – | 0.396(7) | – | 0.391(6) | – | 0.393(6) | – | 0.422(4) | – | ◇ | – | **0.351(6)** |
| UMPM | – | 0.438(5) | – | 0.644(5) | – | 0.368(5) | – | 0.403(4) | – | 0.817 | – | **0.346(5)** |

TABLE II

COMPARISON OF ROTATION ESTIMATION ON BENCHMARK SEQUENCES.

| Methods | PTA($K$) | CSF($K$) | RIKA($K$) | BMM($K$) | SFLBFGS($K$) |
|---|---|---|---|---|---|
| Drink | **0.005(11)** | 0.006(12) | 0.006(12) | 0.007(12) | **0.005(12)** |
| Pick-up | 0.154(8) | 0.155(13) | 0.155(13) | 0.114(13) | **0.105(13)** |
| Yoga | 0.105(13) | 0.102(13) | 0.102(13) | 0.088(10) | **0.085(13)** |
| Stretch | **0.049(13)** | **0.049(13)** | **0.049(13)** | 0.068(11) | **0.049(13)** |

TABLE III

COMPARISON OF 3D SHAPE RECOVERY ON BENCHMARK SEQUENCES.

| Methods | PTA($K$) | CSF($K$) | RIKA($K$) | BMM($K$) | SFLBFGS($K$) |
|---|---|---|---|---|---|
| Drink | 0.023(13) | **0.013(12)** | 0.021(12) | 0.023(13) | **0.013(13)** |
| Pick-up | 0.077(10) | 0.038(10) | 0.053(11) | 0.049(13) | **0.032(13)** |
| Yoga | 0.038(9) | 0.035(7) | 0.037(11) | 0.033(12) | **0.029(13)** |
| Stretch | 0.042(11) | 0.045(8) | 0.043(9) | 0.044(13) | **0.037(13)** |

mid-scale Drink sequence, the large-scale Flag and UMPM sequences, are listed in Table IV.

As seen from Table I and Table IV, three points are revealed: (i) When handling the mid-scale Drink sequence, SFLBFGS runs faster than RIKA and BMM, but more slowly than PTA and CSF. However, when handling the large-scale Flag sequence with a large number of point tracks, SFLBFGS runs faster than CSF, RIKA, BMM(also EM-PND), and just more slowly than PTA. Moreover, although the total number of the matrix entries for the Flag sequence is much larger than the one for the Drink sequence, SFLBFGS runs faster on the Flag sequence than on the Drink sequence, because the computational complexity of Algorithm 1 is strongly related to the frame number and the selected rank rather than the point track number as indicated in Section II-C. In fact, compared with the Drink sequence, both the frame number of the Flag sequence and the used rank for handling the Flag sequence are lower, hence, SFLBFGS runs faster on the Flag sequence than on the Drink sequence. (ii) SFLBFGS runs more slowly than PTA and EM-PND on the UMPM sequence with a large number of frames, but faster than CSF, RIKA, and BMM. Moreover, comparing with the running times of the referred methods on the Drink sequence, those on the UMPM sequence increase at different levels. However, the increasing ranges of SFLBFGS are lower than those of the rest methods. (iii) SFLBFGS obtains better reconstructed shapes than the rest methods on the two large-scale sequences, and achieves an effective tradeoff between shape recovery accuracy and computational speed. This demonstrates that SFLBFGS is capable to handle large-scale data effectively.

TABLE IV

RUNNING TIMES(SECONDS) ON THE MID-SCALE AND LARGE-SCALE DATA.

| Methods | PTA | CSF | RIKA | BMM | EM-PND | SFLBFGS |
|---|---|---|---|---|---|---|
| Drink | 18 | 17 | 2123 | 719 | □ | 390 |
| Flag | 8 | 331 | 1436 | 1525 | ◇ | 82 |
| UMPM | 1228 | 21656 | 1164857 | 33564 | 781 | 13171 |

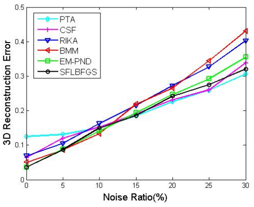


Fig. 1. Performance comparison on noisy Face1 dataset.

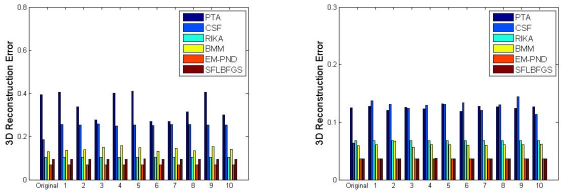
### B. Performances on Noisy Data

This subsection evaluates the performances of all the referred methods on noisy data. Zero mean Gaussian random noise with standard deviations $\sigma_n = r * \max(\text{std}(W_x), \text{std}(W_y))$ ($r \in \{5\%, 10\%, 15\%, 20\%, 25\%, 30\%\}$ is the noise ratio, $W_x/W_y$ represent all the $x/y$-coordinates in $W$) is added into each point of the used standard sequences in Section III-A, and each method is implemented 10 times independently on each sequence with noise.

Fig. 1 shows the performance comparison on the Face1 sequence with different noise ratios. As is seen, the extra-prior-free SFLBFGS performs better than RIKA, BMM, and EM-PND in most cases, and achieves close performances to both PTA and CSF which require extra smoothness priors. This demonstrates that SFLBFGS can effectively handle noisy data.

### C. Performances on Frame-Reshuffled Data

In some special cases, the temporal relations among the input 2D points may be unknown and the temporal-smoothness prior does not hold, such as the case where the input 2D points are obtained from a collection of images without temporal relations. Here, the performances of the referred methods on data without temporal relations are evaluated. Such kind of data



(a) Walking

(b) Face1

Fig. 2. Performance comparison on frame-reshuffled data.

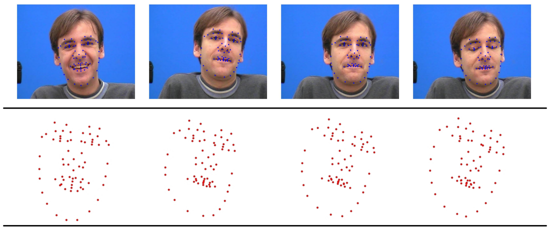


Fig. 3. Performance on real video Data. First row: sample frames and 2D tracking points; Second row: approximately front views of the reconstructed 3D shapes by the proposed method.

is synthesized by randomly reshuffling the frames of the used standard sequences in Section III-A as done in [11], [12], [15], and the smoothness assumption across frames does not hold any more in these frame-reshuffled sequences. These methods are implemented on ten sets of frame-reshuffled data from each standard sequence, and Fig. 2 shows their performances on Walking and Face1. As is seen, EM-PND, RIKA, BMM, SFLBFGS are immune to the random permutation, while PTA, CSF are sensitive to the random permutation.

### D. Performances on Real Video Data

In this section, the Franck sequence[1], which is taken from a video of a person engaged in conversation, is used to evaluate the performance of the proposed SFLBFGS method on real video data as done in [8], [11]. We select the first 1500 frames from the 5000-frame video sequence for testing. An AAM(Active Appearance Model) has been employed to track 68 features in this sequence. Fig. 3 shows four reconstruction samples by SFLBFGS. As is seen, SFLBFGS can effectively recover the 3D shapes from the input real video data.

## IV. Conclusion

In this paper, we propose an extra-prior-free factorization method SFLBFGS for NRSFM, which computes the rotation matrix and the 3D deformable shapes in a sequential manner. SFLBFGS is easy to implement, where only several simple techniques are employed. Moreover, it is able to handle large-scale data and achieve an effective tradeoff between computational speed and shape recovery accuracy as demonstrated in our experiments. In the future, we will investigate how to extend it to handle missing-data cases effectively.

**Acknowledgment**

This work was supported by National Natural Science Foundation of China under grant Nos.(61573359, 61379081, 61375042) and Beijing Natural Science Foundation under grant No.(7142152).

## References

[1] Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: CVPR. (2000) 690–696

[2] Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. PAMI **30**(5) (2008) 878–892

[3] Angst, R., Pollefeys, M.: A unified view on deformable shape factorizations. In: ECCV (6). (2012) 682–695

[4] Tao, L., Matuszewski, B.J.: Non-rigid structure from motion with diffusion maps prior. In: CVPR. (2013) 1530–1537

[5] Garg, R., Roussos, A., de Agapito, L.: Dense variational reconstruction of non-rigid surfaces from monocular video. In: CVPR. (2013) 1272–1279

[6] Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S.I., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: CVPR. (2008)

[7] Paladini, M., Del Bue, A., Xavier, J., Agapito, L., Stošic, M., Dodig, M.: Optimal metric projections for deformable and articulated structure-from-motion. IJCV **96** (2012) 252–276

[8] Del Bue, A., Xavier, J., Agapito, L., Paladini, M.: Bilinear modeling via augmented lagrange multipliers (balm). PAMI **34**(8) (2012) 1496 –1508

[9] Cabral, R.S., la Torre, F.D., Costeira, J.P., Bernardino, A.: Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In: ICCV. (2013) 2488–2495

[10] Zhu, Y., Huang, D., De la Torre Frade, F., Lucey, S.: Complex non-rigid motion 3d reconstruction by union of subspaces. In: CVPR. (2014)

[11] Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. In: CVPR. (2012) 2018–2025

[12] Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. IJCV **107**(2) (2014) 101–122

[13] Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Nonrigid structure from motion in trajectory space. In: NIPS. (2008) 41–48

[14] Gotardo, P.F.U., Martínez, A.M.: Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. PAMI **33**(10) (2011) 2051–2065

[15] Hamsici, O.C., Gotardo, P.F.U., Martínez, A.M.: Learning spatially-smooth mappings in non-rigid structure from motion. In: ECCV (4). (2012) 260–273

[16] Lee, M., Cho, J., Choi, C., Oh, S.: Procrustean normal distribution for non-rigid structure from motion. In: CVPR. (2013) 1280–1287

[17] Cho, J., Lee, M., Oh, S.: Complex non-rigid 3d shape recovery using a procrustean normal distribution mixture model. IJCV (2015)

[18] Jain, P., Meka, R., Dhillon, I.S.: Guaranteed rank minimization via singular value projection. In: NIPS. (2010) 937–945

[19] Skajaa, A.: Limited memory BFGS for nonsmooth optimization. Master's thesis, New York University (2010)

[20] Kulis, B., Surendran, A., Platt, J.: Fast low-rank semidefinite programming for embedding and clustering. In: AISTATS. (2007) 235–242

[21] Akhter, I., Sheikh, Y., Khan, S.: In defense of orthonormality constraints for nonrigid structure from motion. In: CVPR. (2009) 1534–1541

[22] Nocedal, J., Wright, S.J.: Numerical Optimization. 2nd edn. Springer, New York (2006)

[23] Paladini, M., Bue, A.D., Stosic, M., Dodig, M., Xavier, J.M.F., de Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: CVPR. (2009) 2898–2905

[24] Gotardo, P.F.U., Martínez, A.M.: Kernel non-rigid structure from motion. In: Proc. ICCV, Barcelona, Spain (2011) 802–809

[25] Garg, R., Roussos, A., Agapito, L.: Robust trajectory-space TV-$l_1$ optical flow for non-rigid sequences. In: International Conference on Energy Minimization Methods in CVPR. (2011) 300–314

[26] van der Aa, N., Luo, X., Giezeman, G.J., Tan, R., Veltkamp, R.: Utrecht Multi-Person Motion (UMPM) benchmark: a multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In: HICV(in conjunction with ICCV). (2011) 1264–1269

[1] http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face .html