Abstract

This paper addresses the task of dense non-rigid structure-from-motion (NRSfM) using multiple images. State-of-the-art methods to this problem are often hurdled by scalability, expensive computations, and noisy measurements. Further, recent methods to NRSfM usually either assume a small number of sparse feature points or ignore local non-linearities of shape deformations, and thus cannot reliably model complex non-rigid deformations. To address these issues, in this paper, we propose a new approach for dense NRSfM by modeling the problem on a Grassmann manifold. Specifically, we assume the complex non-rigid deformations lie on a union of local linear subspaces both spatially and temporally. This naturally allows for a compact representation of the complex non-rigid deformation over frames. We provide experimental results on several synthetic and real benchmark datasets. The procured results clearly demonstrate that our method, apart from being scalable and more accurate than state-of-the-art methods, is also more robust to noise and generalizes to highly non-linear deformations.

1. Introduction

Non-rigid structure-from-motion (NRSfM) is a classical problem in computer vision, where the task is to recover the 3D shape of a deforming object from multiple images. Despite the fact that NRSfM for arbitrary deformation still remains an open problem, it can be solved efficiently under some mild prior assumptions about the deformation and the shape configuration [14, 28, 23, 41, 4, 24, 25, 27, 26].

Even though the existing solutions to sparse NRSfM have demonstrated outstanding results, they do not scale to dense feature points and their resilience to noise remains unsatisfactory. Moreover, the state-of-the-art algorithms [18, 11] to solve dense NRSfM are computationally expensive and rely on the assumption of global low-rank shape which, unfortunately, fails to cater to the inherent local structure of the deforming shape over time. Consequently, to represent dense non-rigid structure under such formulations seem rather flimsy and implausible.

For many real-world applications, such as facial expression reconstruction, limitations such as scalability, timing, robustness, reliable modeling, etc., are of crucial concern. Despite these limitations, no template-free approach exists that can reliably deal with these concerns. In this paper, we propose a template-free dense NRSfM algorithm that overcomes these difficulties. As a first step to overcome these difficulties, we reduce the overall high-dimensional non-linear space spanned by representing the deforming shape as a union of several local low-dimensional linear subspaces. Our approach is based on a simple idea/assumption i.e., any complex deforming surface can be approximated by a locally linear subspace structure [10]. We use this simple intuition in a spatio-temporal framework to solve dense NRSfM. This choice naturally leads to a few legitimate queries:

allows the same idea as proposed by Kumar et al. [23] has also observed an improvement in the reconstruction accuracy under such formulations. Even though the concept behind such a framework is elementary, no algorithm to our knowledge exists that exploit such an intrinsic idea for dense NRSfM.

b) Why the previously proposed spatio-temporal methods are unable to handle dense NRSfM? The formulation proposed by Kumar et al. [23] and its adaptation [3] is inspired from SSC [16], and LRR [30]. As a result, the complexity of their formulations grows exponentially in the order of the number of data points. This makes it difficult to solve dense NRSfM using their formulations. Moreover, these methods [23, 41, 3] use an assumption that non-rigid shape should lie on a low-dimensional linear or affine subspace globally. In reality, such an assumption does not hold for all kinds of non-linear deformations [39, 34]. Although a recent spatio-temporal method proposed by Dai et al. [11] solves this task, it involves a series of least square problems to be solved, which is computationally demanding.

To overcome all these issues, we propose a spatio-temporal dense NRSfM algorithm which is free from such unwarranted assumptions and limitations. Instead, we adhere to the assumption that the low-dimensional linear subspace spanned by a deforming shape is locally valid. Such an assumption about shapes has been well studied in topological manifold theory [1, 15]. The Grassmann manifold is a topologically rich non-linear manifold, each point of which represents the set of all right-invariant subspaces of the Euclidean space. One property of the Grassmannian that is particularly useful in our setting is that the points in it can be embedded into the space of symmetric matrices. This property has been used in several computer vision applications that deal with subspace representation of data [20, 8]. Accordingly, in our problem, to model a non-linear shape, using a Grassmannian allows us to represent the shape as a set of “smooth” low-dimensional surfaces embedded in a higher dimensional Euclidean space. Such a representation not only reduces the complexity of our task but also makes our formulation robust and scalable as described below.

c) Why Grassmann manifold? It is well-known that the complex non-rigid deformations are composed of multiple subspaces that quite often fit a higher-order parametric model [33, 36, 41]. To handle such complex models globally can be very challenging – both numerically and computationally. Consequently, for an appropriate representation of such a model, we decompose the original non-linearity of the shape by a set of locally linear models that span a low-rank subspace of a vector space. As alluded to above, the space of all $d$-dimensional linear subspaces of $\mathbb{R}^N$ ($0 < d < N$) forms the Grassmann manifold [1, 2]. Modeling the deformation on this manifold allows us to operate on the number of subspaces rather than on the number of vectorial data points (on the shape), which reduces the complexity of the problem significantly. Moreover, since each local surface is a low-rank subspace, it can be faithfully reconstructed using a few eigenvalues and corresponding eigenvectors, which makes such a representation scalable and robust to noise.

The aforementioned properties of the Grassmannian perfectly fit our strategy to model complex deformations, and therefore, we blend the concept of spatio-temporal representations with local low-rank linear models. This idea results in a two-stage coupled optimization problem i.e., local reconstruction and global grouping, which is solved efficiently using the standard ADMM algorithm [7]. As the local reconstructions are performed using a low-rank eigen decomposition, our representation is computationally efficient and robust to noise. We demonstrate the benefit of our approach to benchmark real and synthetic sequences §3. Our results show that our method outperforms previous state-of-the-art approaches by 1-2% on all the benchmark datasets. Before we provide the details of our algorithm, we review some pertinent previous works in the next section.

2. Background

This section provides a brief background on the recent advancements in NRSfM, focusing mainly on the methods that are relevant to this work.

Preliminaries: We borrow the notation system from Dai et al.’s work [14] for its wide usage. Given $‘P’$ feature points over $‘F’$ frames, we represent $W \in \mathbb{R}^{2F \times P}$, $S \in \mathbb{R}^{3F \times P}$, $R \in \mathbb{R}^{2F \times 3F}$ as the measurement, the shape, and the rotation matrices, respectively. Here $R$ matrix is composed of block diagonal $R_i \in \mathbb{R}^{2 \times 3}$, representing per frame orthographic camera projection. Also, the notation $S^\# \in \mathbb{R}^{M' \times F}$ stands for the rearranged shape matrix, which is a linear mapping of $S$. We use $\| : \|_F$ and $\| : \|_*$ to denote the Frobenius norm and the nuclear norm, respectively.

2.1. Relevant Previous Work

Dai et al.’s approach: Dai et al. proposed a simple and elegant solution to NRSfM [14]. The work, dubbed “prior-free”, provides a practical solution as well as new theoretical insights to NRSfM. Their formulation involves nuclear norm minimization on $S^\#$ instead of $S$ –see Table 1(a). This is enforced due to the fact that $3K$ rank bound on $S$ is weaker than $K$ rank bound on $S^\#$, where $K$ refers to the rank of $S$. Although this elegant framework provides robust results for the shapes that span a single subspace, it may perform poorly on complex non-rigid motions [41].

Zhu et al.’s approach: To achieve better 3D reconstructions on complex non-rigid sequences, this work capitalized on the limitations of Dai et al.’s work [14] by exploiting the union of subspaces in the shape space [41]. The proposed formulation is inspired by LRR [30] in conjunction with Dai
et al. work –see Table 1(b). In the formulation, $C \in \mathbb{R}^{F \times F}$, $E \in \mathbb{R}^{2F \times F}$ are the coefficient and error matrices.

Kumar et al.’s approach: Kumar et al. exploits multiple subspaces both in the trajectory space and in the shape space [23]. This work demonstrated empirically that procuring multiple subspaces in the trajectory and shape spaces provide better reconstruction results. They proposed a joint segmentation and reconstruction framework, where segmentation inherently benefits reconstruction and vice-versa –see Table 1(c). In their formulation $C_1 \in \mathbb{R}^{P_1 \times P}$, $C_2 \in \mathbb{R}^{P_2 \times F}$ are the coefficient matrices and, $g(.)$ linearly maps $S$ to $S^t$.

Dense NRSfM approach: Garg et al. developed a variational approach to solve dense NRSfM [18]. The optimization framework proposed by them employs total variational constraint on the deforming shape ($\nabla S^t_j(p)$) to allow edge preserving discontinuities, and trace norm constraints to penalize the number of independent shapes –see Table 1(d). Recently, Dai et al. has also proposed a dense NRSfM algorithm with a spatio-temporal formulation [11].

2.2. Motivation

This work is intended to overcome the shortcomings of the previous approaches to solve dense NRSfM. Accordingly, we would like to outline the critical limitations associated with them. Although some of them are highlighted before, we reiterate it for the sake of completeness.

(a) To solve dense NRSfM using the formulation proposed by Kumar et al. [23] and Zhu et al. [41] is nearly impossible due to the complexity of their formulation §1. Also, the error measure used by them is composed of Euclidean norm defined on the original data (see Table 1), which is not proper for non-linear data with a manifold structure [1, 38].

(b) The algorithm proposed by Garg et al. [18] results in a biconvex formulation, which is computationally expensive and needs a GPU to speed up the implementation. Similarly, Dai et al.’s recent work [11] is computationally expensive as well due to costly gradient term in their formulation.

(c) Methods such as [40, 29] rely on the template prior for dense 3D reconstruction of the object. Other piecewise approach for solving dense NRSfM [35] require a post-processing step to stitch all the local reconstructions.

To avoid all the aforementioned limitations, we propose a new dense NRSfM algorithm. The primary contributions of this paper are as follows:

1. A scalable spatio-temporal framework on the Grassmann manifold to solve dense NRSfM which does not need any template prior.
2. An effective framework that can handle non-linear deformations even with noisy trajectories and provides state-of-the-art results on benchmark datasets.
3. An efficient solution to the proposed optimization based on the ADMM procedure [7].

Table 1: A brief summary of formulation used by some of the recent approaches to solve sparse and dense NRSfM which are closely related to our method. Among all these four methods only Garg et al.‘s [18] approach is formulated particularly for solving dense NRSfM.

<table>
<thead>
<tr>
<th>Method</th>
<th>Formulation</th>
</tr>
</thead>
</table>
| (a) Dai et al.’s [14] | \[
\begin{align*}
\text{minimize} & \quad \|S^t + \lambda |E|\|_F^2, \\
\text{subject to:} & \quad W = RS + E
\end{align*}
\] |
| (b) Zhu et al.’s [41] | \[
\begin{align*}
\text{minimize} & \quad \|C\|_* + \gamma \|S^t\|_* + \lambda |E|_1 \\
\text{subject to:} & \quad S^t = SC, W = RS + E
\end{align*}
\] |
| (c) Kumar et al.’s [23] | \[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}\|W - RS\|_F^2 + \lambda_1 \|C_1\|_1 + \lambda_2 \|S^t\|_* + \lambda_3 \|C_2\|_1 \\
\text{subject to:} & \quad S = SC_1, S^t = C_2, T^T C_1 = T^T, T^T C_2 = T^T, \\
& \quad \text{diag}(C_1) = 0, \text{diag}(C_2) = 0, S^t = g(S)
\end{align*}
\] |
| (d) Garg et al.’s [18] | \[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}\|W - RS\|_F^2 + \sum_{i,p} \|\nabla S^t_j(p)\|_* + \tau \|S^t\|_* \\
\text{subject to:} & \quad R \in \mathcal{SO}(3)
\end{align*}
\] |

3. Problem Formulation

In this section, we first provide a brief introduction to the Grassmann manifold and a suitable definition for a similarity distance metric on it, before revealing our formulation.

3.1. Grassmann Manifold

The Grassmann manifold, usually denoted as $\mathcal{G}(n, r)$, consists of all $r$-dimensional linear subspaces of $\mathbb{R}^n$, where $n > r$. A point on the Grassmann manifold is represented by a $n \times r$ matrix (say $X$), whose columns are composed of orthonormal basis of the subspace spanned by $X$, denoted as $\text{span}(X)$ or in short as $[X]$. Let’s suppose $[X_1], [X_2]$ are two such points on this manifold, then among several similarity distances known for this manifold [20], we will be using the projection metric distance given by $d_g([X_1], [X_2]) = \frac{1}{\sqrt{2}} \|X_1X_1^T - X_2X_2^T\|_F$, as it allows directly embedding the Grassmannian points into a Euclidean space (and the use of the Frobenius norm) using the mapping $X \rightarrow XX^T$. With this metric, $(\mathcal{G}, d_g)$ forms a metric space. Interested readers may refer to [20] for details.

3.2. Formulation

With the relevant background as reviewed in the above sections, we are now ready to present our algorithm to solve the dense NRSfM task under orthographic projection. We start our discussion with the classical representation to NRSfM i.e.,

\[ W_s = RS_s \]
\[ W_s \in \mathbb{R}^{2F \times P}, \quad R = \text{blkdag}(R_1, ..., R_F) \in \mathbb{R}^{2F \times 3F}, \quad S_s \in \mathbb{R}^{3F \times P}. \]

The motive here is, given the input measurement matrix, solve for rotation \((R)\) and 3D shape \((S_s)\). To serve this objective, Eq.(1) maintains the camera motion and the shape deformation such that it complies with the image measurements. For our method, we solve for rotations using the Intersection method \([14]\) by assuming that the multiple non-rigid motions within a single deforming object, over frames, can be faithfully approximated by per frame single relative camera motion with a higher rank\(^1\). Accordingly, our goal reduces to develop a systematic approach that can reliably explain the non-rigid shape deformations and provides better 3D reconstruction. We use subscript ‘s’ in Eq.(1) to indicate that the column permutations of \(S_s\) and \(W_s\) matrix are allowed. However, the column permutations of \(S_s^T\) is inadmissible as it results in discontinuous trajectories over frames.

**Grassmannian Representations in Trajectory Space:**

Let’s suppose \(\Psi_s = \{\psi_{s1}, \psi_{s2}, ..., \psi_{sK_s}\}\) is the set of points on the Grassmann manifold generated using \(S_s\) matrix, then \(\mathcal{T}_s = \{\psi_{s1}(\psi_{s1})^T, (\psi_{s2})(\psi_{s2})^T, ..., (\psi_{sK_s})(\psi_{sK_s})^T\}\) represents a tensor which is constructed by mapping all symmetric matrices of the Grassmann data points — refer Figure 2(a). As discussed before in §1, to explain the complex deformations, we reduce the overall non-linear space as a union of several local low-dimensional linear spaces which form the sample points on the Grassmann manifold. But, the notion of self-expressiveness is valid only for Euclidean linear or affine subspace. To apply self-expressiveness on the Grassmann manifold one has to adopt linearity onto the manifold. Since, Grassmann manifold is isometrically equivalent to the symmetric idempotent matrices \([9]\), we embed the Grassmann manifold into the symmetric matrix manifold, where the self-expression can be defined in the embedding space. This leads to the following optimization:

\[
\begin{aligned}
&\minimize_{E_s, C_s} \|E_s\|_F^2 + \lambda_1 \|C_s\|_*, \\
&\text{subject to: } T_s = T_sC_s + E_s.
\end{aligned}
\]  

(2)

We denote \(C_s \in \mathbb{R}^{K_s \times K_s}\) as the coefficient matrix with ‘\(K_s\)’ as the total number of spatial groups. Here, \(E_s\) measures the trajectory group reconstruction error as per the manifold geometry. Also, we would like to emphasize that since the object undergoes deformations in the 3D space, we operate in 3D space rather than in the projected 2D space. \(\| \|_s\) is enforced on \(C_s\) for a low-rank solution.

**Grassmannian Representations in Shape Space:**

Deforming object attains different state over time which adheres to distinct temporal local subspaces \([23]\). Assuming that the temporal deformation is smooth over-time, we express deforming shapes in terms of local self-expressiveness across frames as:

\[
\begin{aligned}
&\minimize_{E_t, C_t} \|E_t\|_F^2 + \lambda_2 \|C_t\|_*, \\
&\text{subject to: } T_t = T_tC_t + E_t.
\end{aligned}
\]  

(3)

Similarly, \(T_t\) is the set of all symmetric matrices constructed using a set of Grassmannian samples \(\Psi_t\), where \(\Psi_t\) contains the samples which are drawn from \(S_t^2 \in \mathbb{R}^{3P \times F}\) — refer Figure 2(b). Intuitively, \(S_t^2\) is a shape matrix with each column as a deforming shape. \(E_t, C_t \in \mathbb{R}^{K_t \times K_t}\) represent the temporal group reconstruction error and coefficient matrix respectively, with \(K_t\) as the number of temporal groups. \(\| \|_t\) is enforced on \(C_t\) for a low-rank solution.

**Spatio-Temporal Formulation:** Combining the above two objectives and their constraints with reprojection error term give us our formulation. Our representation blends the local subspaces structure along with the global composition of a
non-rigid shape. Thus, the overall objective is:

\[
\begin{align*}
\text{minimize} & \quad E = \frac{1}{2} \|W_s - RS_s\|_F^2 + \gamma \|S_t\|_* + \lambda_1 \|E_s\|_F^2 \\
& + \lambda_2 \|E_t\|_F^2 + \lambda_3 \|C_s\|_* + \lambda_4 \|C_t\|_* \\
\text{subject to:} & \quad T_s = T_sC_s + E_s; \quad T_t = T_tC_t + E_t; \\
& \quad \Psi_s = \xi(C_s, S_s, \sigma_c); \quad \Psi_t = \xi(C_t, S_t^1, \sigma_c); \\
& \quad S_s = \psi(\Psi_s, \Sigma_s, V_s, N_s); \quad S_t^1 = \psi(\Psi_t, \Sigma_t^1, V_t, N_t); \\
& \quad S_t^2 = \mathcal{F}(S_s); \quad W_s = \mathcal{F}(W_s, S_s);
\end{align*}
\]

The re-projection error constraint performs the 3D reconstruction using \( W_s \) and \( R \). Meanwhile, the local subspace grouping naturally enforces the union of subspace structure in \( S_s, S_t^2 \) with corresponding low-rank representations of the coefficient matrices \( C_s \) and \( C_t \). Here, the function \( \xi(\cdot) \) draws inference from \( C \) matrices to refine Grassmannian sample set, both in trajectory and shape spaces. The function \( \psi(\cdot) \) reconstruits \( S_s \) and \( S_t^1 \) matrices based on a set of local subspaces \( (\Psi_s, \Psi_t, V_s, V_t) \), singular values \( (\Sigma_s, \Sigma_t) \) and the number of top eigenvalues \( (N_s, N_t) \). The function \( \mathcal{F}(\cdot) \) transforms \( S_s \in \mathbb{R}^{M \times F} \) matrix to \( S_t^2 \in \mathbb{R}^{M \times F} \) matrix and \( \mathcal{F}(\cdot) \) function rearranges \( W_s \) matrix as per the recent ordering of \( S_s^2 \). Parameters such as ‘\( \sigma_c \)’, ‘\( N_s \)’ and ‘\( N_t \)’ provides the flexibility to handle noise and adjust computations. Note that the element of the sets \( \Psi_s, \Psi_t, V_s \) and \( V_t \) are obtained using SVD. The above equation i.e. Eq: (4) is a coupled optimization problem where the solution to \( S \) matrices influence the solution of \( C \) matrices and vice-versa, and \( \mathcal{F}(\cdot) \) connects \( S_t^2 \) to \( S_s \).

4. Solution

The formulation in Eq.(4) is a non-convex problem due to the bilinear optimization variables \( (T_sC_s, T_tC_t) \), hence a global optimal solution is hard to achieve. However, it can be efficiently solved using Augmented Lagrangian Methods (ALMs) [7], which has proven its effectiveness for many non-convex problems. Introducing Lagrange multipliers \((\{Y_i\}_{i=1}^3)\) and auxiliary variables \((J_s, J_t)\) to Eq.(4) gives us the complete cost function as follows:

\[
\begin{align*}
\text{minimize} & \quad E = \frac{1}{2} \|W_s - RS_s\|_F^2 + \frac{3}{2} \|S_t^1 - \mathcal{F}(S_s)\|_F^2 + \\
& + \lambda_1 \|T_sC_s\|_F^2 + \lambda_3 \|C_s\|_* + \lambda_4 \|C_t\|_* \\
& + \lambda_2 \|C_s - J_s\|_F^2 + \lambda_3 \|C_t - J_t\|_F^2 + < Y_s, C_s - J_s > + < Y_t, C_t - J_t > \\
& + \lambda_1 \|J_s\|_* + \lambda_4 \|J_t\|_* \\
\text{subject to:} & \quad \Psi_s = \xi(C_s, S_s, \sigma_c); \quad \Psi_t = \xi(C_t, S_t^1, \sigma_c); \\
& \quad S_s = \psi(\Psi_s, \Sigma_s, V_s, N_s); \quad S_t^1 = \psi(\Psi_t, \Sigma_t^1, V_t, N_t); \\
& \quad W_s = \mathcal{F}(W_s, S_s);
\end{align*}
\]

The function \( \xi(\cdot) \) first computes the SVD of \( C \) matrices, i.e. \( C = [U_c, \Sigma_c, V_c] \), then forms a matrix \( A \) such that \( A_{ij} = [X^{T}T_{ij}^{T}]_t \), where \( \sigma_t \) is set empirically based on noise levels and \( X = U_c(\Sigma_c)^{0.5} \) (normalized). Secondly, it uses \( A_{ij} \) to form new Grassmann samples from the \( S \) matrices [22]. Notice that \( \xi(\cdot) \) operates on \( C \) matrices whose dimensions depend on the number of Grassmann samples. This reduces the complexity of the task from exponential in the number of vectorial points to exponential in the number of linear subspaces. The later being of the order 10-50, where as the former can go more than 50,000 for dense NRSIM.

The \( \xi(\cdot) \) function is defined as follows \( \zeta = \{ (\Psi_a, \Sigma_a, V_a, r), S_a = \text{horzcat}(\Psi_a, \Sigma_a^0 V_a^T), \forall 1 \leq a \leq \text{Card}(\Psi_a), r \in \mathbb{Z}^+ \} \), where \( r \) stands for top-\( r \) eigenvalues, \( \text{Card}(\cdot) \) denotes the cardinal number of the set and \( \text{horzcat}(\cdot) \) denotes for the horizontal concatenation of matrices. Intuitively, \( \xi(\cdot) \) reconstructs back each local low-rank subspace. During implementation, we use \( S_s, S_t^2 \) in place of \( S_s \) accordingly. The optimization variables over iteration are obtained by solving for one variable at a time treating others as constant, keeping the constraints intact. For detailed derivations for each sub-problem and proofs, kindly refer to the supplementary material. The pseudo code of our implementation is provided in Algorithm 1.

5. Experiments and Results

We compare the performance of our method against four previously reported state-of-the-art approaches, namely Dense Spatio-Temporal DS [11], Dense Variational DV [18], Trajectory Basis PTA [4] and Metric Projection MP [32]. To test the performance, we used dense NRSIM dataset introduced by Garg et al. [18] and Varol et al. [37] under noisy and noise free conditions. For quantitative evaluation of 3D reconstruction, we align the estimated shape \( S_{est}^t \) with ground-truth shape \( S_{GT}^t \) per frame using Procrustes analysis. We compute the average RMS 3D reconstruction error as \( e_{3D} = \frac{1}{T} \sum_{t=1}^{T} \|S_{est}^t - S_{GT}^t\|_F \). We used Kmeans++ algorithm [6] to initialize segments without disturbing the temporal continuity.

Experiments on Synthetic Face Sequences: This dataset consists of 4 different face sequence with 28,880 feature points tracked over multiple frames. The face sequence 1, 2 is a 10 frame long video, whereas, face sequence 3, 4 is a 99 frame long video. It’s a challenging dataset mainly due to different rotation frequencies and deformations in each of the sequence. Figure 3 shows the qualitative reconstruction results obtained using our approach in comparison to the ground-truth for face sequence 4. Table 2 lists the performance comparisons of our method with other competing methods. Clearly, our algorithm outperforms the other baseline approach, which helps us to conclude that holistic approaches to rank minimization without drawing any
**Algorithm 1 Scalable Dense Non-Rigid Structure from Motion: A Grassmannian Perspective**

**Require:** $W_s$, $R$ using [14], tuning parameters: $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \gamma, \rho = 1.1, \beta = 1e^{-3}, \beta_m = 1e^0, \epsilon = 1e^{-12}, K_s, K_f$.

**Initialize:** $S_s$ = pseudoinverse($R$)$W_s$ and $S_t^O = \mathcal{S}_t^O(S_s)$.

**Initialize:** ‘$K_f$’ temporal data points on the Grassmann manifold using ‘$S_t^O$’ matrix, $\Psi_t = \{ \Psi_{ts} \}_{i=1}^{K_t}$.

**Initialize:** ‘$S_s$’ spatial data points on the Grassmann manifold using ‘$S_t^O$’ matrix, $\Psi_s = \{ \Psi_{ss} \}_{i=1}^{K_s}$.

**Initialize:** The auxiliary variables $J_s$, $J_t$ and Lagrange multipliers $\{ Y_i \}_{i=1}^3$ as zero matrices.

**Initialize:**

$\Omega_{ij}^s = \text{trace}([ \Psi_{ij}^s \Psi_{ij}^s ] (\Psi_{ij}^s \Psi_{ij}^s)), \Omega_{ij}^t = \text{trace}([ \Psi_{ij}^t \Psi_{ij}^t ] (\Psi_{ij}^t \Psi_{ij}^t)), \Omega_s = (\Omega_{ij}^s)_{i,j=1}^{K_s}, \Omega_t = (\Omega_{ij}^t)_{i,j=1}^{K_t}$

$L_sL_t^T = \text{Cholesky}(\Omega_s), L_tL_t^T = \text{Cholesky}(\Omega_t)$

1: while not converged do

2: $S_s \leftarrow (R^T R + \beta I)^{-1} \left( \beta \left( \mathcal{J}^{-1}_1(S_t^O) + \frac{R^T W_s}{\beta} \right) + R^T W_s \right)$

3: $C_s \leftarrow 2\lambda_1 L_s L_t^T + \beta (J_s - \frac{Y_s}{2}) \left( 2\lambda_1 L_s L_t^T + \beta I \right)^{-1}$

4: $\Psi_s \leftarrow \xi(C_s, S_s, \sigma_s)$ // Update spatial Grassmann points

5: $S_s \leftarrow \zeta(\Psi_s, \Sigma_s, V_s, N_s)$; // refine based on top $N_s$ eigen value

6: $J_s \leftarrow U_s S_s^O (\Sigma_s) V_s$, where $[U_s, \Sigma_s, V_s] = \text{svd}(C_s + \frac{Y_s}{2})$ and $S_s[x] = \text{sign}(x)\max(|x| - \tau, 0)$

7: $S_t^O \leftarrow U_t S_s^O (\Sigma_t) V_t$, where $[U_t, \Sigma_t, V_t] = \text{svd}(C_t + \frac{Y_t}{2})$ and $S_t^O[x] = \text{sign}(x)\max(|x| - \tau, 0)$

8: $C_t \leftarrow 2\lambda_2 L_t L_t^T + \beta (J_t - \frac{Y_t}{2}) \left( 2\lambda_2 L_t L_t^T + \beta I \right)^{-1}$

9: $\Psi_t \leftarrow \xi(C_t, S_t^O, \sigma_t)$ // Update temporal Grassmann points

10: $S_t^O \leftarrow \zeta(\Psi_t, \Sigma_t, V_t, N_t)$; // refine based on top $N_t$ eigen value

11: $J_t \leftarrow U_t S_s^O (\Sigma_t) V_t$, where $[U_t, \Sigma_t, V_t] = \text{svd}(C_t + \frac{Y_t}{2})$ and $S_t^O[x] = \text{sign}(x)\max(|x| - \tau, 0)$

12: $\Omega_{ij}^s \leftarrow \text{trace}([ \Psi_{ij}^s \Psi_{ij}^s ] (\Psi_{ij}^s \Psi_{ij}^s)); \Omega_{ij}^t \leftarrow \text{trace}([ \Psi_{ij}^t \Psi_{ij}^t ] (\Psi_{ij}^t \Psi_{ij}^t));$

13: $\Omega_s \leftarrow (\Omega_{ij}^s)_{i,j=1}^{K_s}, \Omega_t \leftarrow (\Omega_{ij}^t)_{i,j=1}^{K_t};$ // if $\Omega_s \geq 0, \Omega_t \geq 0$, if $\Omega_s||\Omega_t = 0$ add $\delta I$ to make it $> 0$ (see suppl. material)

14: $L_sL_t^T = \text{Cholesky}(\Omega_s), L_tL_t^T = \text{Cholesky}(\Omega_t);$ // Note: Column permutation for $W_s$ and $S_s$ should be same.

15: $W_s \leftarrow \mathcal{S}_2(W_s, S_s)$ // Note: $\delta$ is a very small positive number and $I$ symbolizes identity matrix.

16: $Y_1 := Y_1 + \beta(S_t^O - \mathcal{S}_2(S_s)), Y_2 := Y_2 + \beta(C_s - J_s), Y_3 := Y_3 + \beta(C_t - J_t);$ // Update Lagrange multipliers

17: $\beta \leftarrow \text{minimum}(\rho \beta, \beta_m)$

18: $\text{maxgap} := \text{maximum}(||S_t^O - \mathcal{S}_2(S_s)||_\infty, ||C_s - J_s||_\infty, ||C_t - J_t||_\infty)$

19: if (maxgap < $\epsilon$ || $\beta > \beta_m$) then

20: break;

21: end if // check for the convergence

22: end while // Note: Kindly use economical version of $\text{svd}$ on a regular desktop.

**Ensure:** $S_s, S_t, C_s, C_t.$

---

Figure 3: Reconstruction results obtained on synthetic dense face dataset (face sequence 4). **Top row:** Ground-truth 3D points, **Bottom row:** Recovered 3D points using our approach.
Figure 4: Qualitative reconstruction results procured on benchmark real dense dataset [18] a) Face sequence (28,332 feature points over 120 frames) b) Back sequence (20,561 feature points over 150 frames) c) Heart sequence (68,295 feature points over 80 frames).

Figure 5: Reconstruction results on benchmark kinect paper and kinect tshirt sequence. Top row: Input image frame. Bottom row: Dense 3D reconstruction for the corresponding frame using our approach.

Experiments on face, back and heart sequence: This dataset contains monocular videos of human facial expressions, back deformations, and beating heart under natural lighting conditions. The face sequence, back sequence, and heart sequence are composed of 28,332, 20,561, and 68,295 feature points tracked over 120, 150, and 80 images, respectively. Unfortunately, due to the lack of ground-truth 3D data, we are unable to quantify the performance of these sequences. Figure 4 shows some qualitative results obtained using our algorithm on this real dataset.

Experiments on noisy data: To evaluate the robustness of our method to noise levels, we performed experiments by adding Gaussian noise under different standard deviations to the measurement matrix. Similar to DS [11] the standard deviations are incorporated as $\sigma_n = r \max |W_s|$ by varying $r$ from 0.01 to 0.05. We repeated the experiment 10 times. Figure 6(a) and Figure 6(b) show the variation in the performance of our method under different noise ratio’s on synthetic face sequences [18] and kinect sequences [37] respectively. It can be inferred from the plot that even with large noise ratios, the average reconstruction error does not...

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>paper</td>
<td>0.0612</td>
<td>-</td>
<td>0.0918</td>
<td>0.0827</td>
<td>0.0394</td>
</tr>
<tr>
<td>tshirt</td>
<td>0.0636</td>
<td>-</td>
<td>0.0712</td>
<td>0.0741</td>
<td>0.0362</td>
</tr>
</tbody>
</table>

Table 3: Average 3D reconstruction error ($e_{3D}$) comparison on kinect paper and kinect tshirt sequence. Note: The code for DV [18] is not publicly available. The pixels with no 3D data available were discarded for the experiments and the evaluation.
fluctuate significantly. This improvement is expected from our framework as it is susceptible only to top eigen values.

**Effects of variable initialization on the overall performance:** We performed several other experiments to study the behavior of our algorithm under different variable initializations. For easy exposition, we conducted this experiment on noise-free sequences. We mainly investigated the behavior of $N_s, N_t, K_s, K_t$ on the overall performance of our algorithm. Figure 6(c) and Figure 6(d) shows the variations in the reconstruction errors with respect to $N_s$ and $K_s$ respectively. A similar trend in the plots is observed for changes on $N_t$ and $K_t$ values. These plots clearly illustrate the usefulness of our local low-rank structure i.e., considering a small number of eigenvalues for every local structure is as good as considering all eigenvalues. Similarly, increasing the number of local subspaces after a certain value has negligible effect on the overall reconstruction error. Furthermore, we examined the form of $C_s$ and $C_t$ after convergence as shown Figure 7(a) and Figure 7(b). Unfortunately, due to the lack of ground-truth data on local subspaces, we could not quantify $C_s$ and $C_t$. For qualitative analysis on the observation, kindly refer to the supplementary material.

**Ablation Analysis:** This test is performed to evaluate the importance of spatial and temporal constraints in our formulation. To do this, we observe the performance of our formulation under four different setups: a) without any spatio-temporal constraint (NC), b) with only spatial constraint (SP), c) with only temporal constraint (TP), and d) with spatio-temporal constraint (Both). Figure 7(c) shows the variations in reconstruction errors under these setups on four synthetic face sequence. The statistics clearly illustrate the importance of both constraints in our formulation.

**Runtime Analysis:** This experiment is performed on a computer with an Intel core i7 processor and 16GB RAM. The script to compute the runtime is written in MATLAB 2016b. Figure 7(d) shows the runtime comparisons of our approach with other dense NRSfM methods. The runtime reported in Figure 7(d) corresponds to the results listed in Table 2, 3. The results clearly show the scalability of our method on datasets with more than 50,000 points. Despite PTA [4] is faster than our approach, its accuracy suffers by a large margin for dense NRSfM (see Table 2, 3).

### 6. Conclusion

In this paper, we have introduced a scalable dense NRSfM algorithm which efficiently models the complex non-linear deformations. We achieved this by exploiting the non-linearity on the Grassmann manifold via a spatio-temporal formulation. Moreover, we provided an efficient ADMM [7] based solution for solving our optimization. In the future, we will consider how to extend this work to the projective setting with perspective cameras (e.g. [13, 12]).

**Acknowledgement.** S. Kumar was supported in part by Australian Research Council (ARC) grant (DE140100180). Y. Dai was supported in part by National 1000 Young Talents Plan of China, Natural Science Foundation of China (61420106007), and ARC grant (DE140100180). H. Li is funded in part by ARC Centre of Excellence for Robotic Vision (CE140100016).
References


