Jumping Manifolds: Geometry Aware Dense Non-Rigid Structure from Motion

Suryansh Kumar
Australian National University, CECS, Canberra, ACT, 2601
suryansh.kumar@anu.edu.au

Abstract

Given dense image feature correspondences of a non-rigidly moving object across multiple frames, this paper proposes an algorithm to estimate its 3D shape for each frame. To solve this problem accurately, the recent state-of-the-art algorithm reduces this task to set of local linear subspace reconstruction and clustering problem using Grassmann manifold representation [34]. Unfortunately, their method missed on some of the critical issues associated with the modeling of surface deformations, for e.g., the dependence of a local surface deformation on its neighbors. Furthermore, their representation to group high dimensional data points inevitably introduce the drawbacks of categorizing samples on the high-dimensional Grassmann manifold [32, 31]. Hence, to deal with such limitations with [34], we propose an algorithm that jointly exploits the benefit of high-dimensional Grassmann manifold to perform reconstruction, and its equivalent lower-dimensional representation to infer suitable clusters. To accomplish this, we project each Grassmannians onto a lower-dimensional Grassmann manifold which preserves and respects the deformation of the structure w.r.t its neighbors. These Grassmann points in the lower-dimension then act as a representative for the selection of high-dimensional Grassmann samples to perform each local reconstruction. In practice, our algorithm provides a geometrically efficient way to solve dense NRSfM by switching between manifolds based on its benefit and usage. Experimental results show that the proposed algorithm is very effective in handling noise with reconstruction accuracy as good as or better than the competing methods.

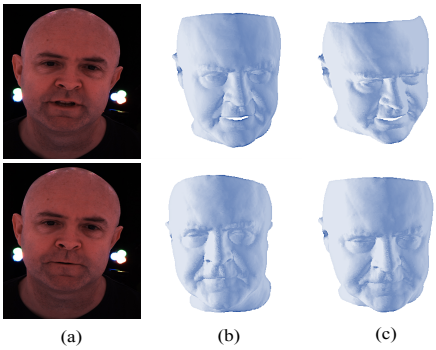


Figure 1: Dense 3D reconstruction of facial expression using our algorithm. The result show the 3D reconstruction of 73,765 points of a complex non-rigidly deforming surface. These results can be useful for real world applications such as 3D modeling, virtual reality etc. The example sequence is taken from Actor dataset [7].

1. Introduction

Non-rigid Structure-from-Motion (NRSfM), a problem where the task is to recover the three-dimensional structure of a deforming object from a set of image feature correspondences across frames. Any solution to this problem depends on the proper modeling of *structure* $\in \mathcal{M}$ and an efficient estimation of *motion* $\in \mathbb{SE}(3)$, where $\mathcal{M}$ denotes some structure manifold and $\mathbb{SE}(3)$ denotes special Euclidean group which is a differentiable manifold [20]. Though, after Bregler *et al*. factorization framework to NRSfM [48], motion estimations are mostly relaxed to rotation estimation $\in \mathbb{SO}(3)$. Even after such relaxation, the problem still remains unsolved for any arbitrary motion. The main difficulty in NRSfM comes from the fact that both the camera and the object are moving and, along with it the object themselves are deforming, hence, it becomes difficult to distinguish camera motion from object motion using only image data. Despite such difficulties, many efficient and reliable solutions based on the priors are proposed to solve NRSfM. A reliable solution to this problem is important as it covers a wide range of applications from medical industry to the entertainment industry and many more.

To solve NRSfM, the algorithms proposed in the past can broadly be divided into two major classes 1) *sparse* NRSfM and 2) *dense* NRSfM. This classification is based on the number of feature points that the algorithm can efficiently process to model the deformation of the object. Although many reliable solution to this problem exists for sparse NRSfM [18, 37, 35, 3, 47, 42, 40, 26, 29], very few work have been done towards solving the dense NRSfM re-

liably and efficiently [23, 17, 34, 4]. Also, the existing solutions to dense NRSfM are computationally expensive and are mostly constrained to analyze the global deformation of the non-rigid shape [23, 17]. The basis for this gradual progress in dense NRSfM is perhaps due to its dependence on per pixel reliable correspondences across frames, and the absence of a resilient structure modeling framework to capture the local non-linearities. One may argue on the efficient motion estimation, however, from image correspondences, we can only estimate relative motion and reliable algorithms with solid theory exists to perform this task well [18, 40]. Also, with the recent progress in deep learning algorithms, per pixel correspondences can be achieved with a remarkable accuracy [46], which leaves structure modeling as a potential gray area in dense NRSfM to focus.

Very recently, Kumar *et al*. [34] has exploited the Grassmann manifold to model non-rigid surfaces in dense NRSfM. The key insight in their work is; even though the overall complexity of the deforming shape is high, each local deformation may be less complex [13, 14, 15, 16]. Using this idea, they proposed a union of local linear subspace approach to solve dense NRSfM problem. Nevertheless, their work overlooked on some of the intrinsic issues associated with the modeling of non-rigidly deforming surface. *Firstly*, their method represents each local linear subspace independently via a high-dimensional Grassmannian representation. Now, such representation may help reconstruct complex 3D deformation but can lead to wrong clustering, and it's very important in joint reconstruction and clustering framework to have suitable clustering of subspaces, else reconstruction may suffer. *Secondly*, their approach to represent local non-linear deformation completely ignored the neighboring surfaces, which may result in an inefficient representation of the Grassmannians in the trajectory space. *Thirdly*, the representation of Grassmannians in the shape space adopted by [34] results in *irredeemable* discontinuity of the trajectories (see Fig.(2)). Hence, temporal representation of the set of shapes using Grassmannians seems not an extremely beneficial choice for modeling dense NRSfM on Grassmannian manifold[1]. *Lastly*, although the dense NRSfM algorithm proposed in [34] works better and faster than the previous methods, it depends on several manual parameters which are inadmissible for practical applications.

This paper introduces an algorithm that overcomes the aforementioned limitations with Kumar *et al*. method [34]. The main point we are trying to make is that; reconstruction and grouping of subspace on the same high dimensional Grassmann manifold seem an unreasonable choice. Even recent research in the Riemannian geometry has shown that the low-dimensional representation of the corresponding high dimensional Grassmann manifold is more favorable

[1]Purpose behind NRSfM is not the same as activity/action recognition. See supplementary material for a detail discussion on this.

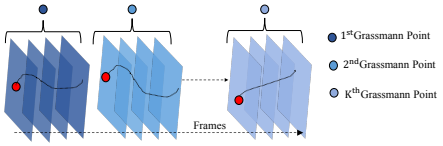


Figure 2: Temporal representation using Grassmannians in the shape space introduces discontinuity in the overall trajectory of the feature point. Also, to define neighboring subspace dependency graph in the time domain seems very challenging keeping in mind that the activity/expression may repeat. Red circle shows the feature point with its trajectory over frames (Black).

for grouping Grassmannians [32, 31]. So, we formulate dense NRSfM in a way that it takes advantage of both high and low dimensional representation of Grassmannians *i.e*., perform reconstruction in the original high-dimension and cluster subspace in its lower-dimension representation.

We devise an unsupervised approach to efficiently represent the high-dimensional non-rigid surface on a lower dimensional Grassmann manifold. These low-dimensional Grassmannians are represented in such a way that it preserves the local structure of the surface deformation in accordance with its neighboring surfaces when projected. Now, these low-dimensional Grassmannians serves as a potential representative for its high-dimensional Grassmannians for suitable grouping, which subsequently help improve the reconstruction and representation of the Grassmannians on the high-dimensional Grassmann manifold, hence, the term **J***umping* **M***anifolds* (**M**o**J**u). Further, we drop the temporal grouping of shapes using Grassmannians to discourage the discontinuity of trajectories (see Fig.(2)).

In essence, our work is inspired by [34] and is oriented towards settling its important limitations. Moreover, in contrast to [34], we capture the notion of dependent local subspace in a union of subspace algorithm [39] via Grassmannian modeling. The algorithm we proposed is an attempt to supply a more efficient, reliable and practical solution to this problem. Our formulation gives an efficient framework for modeling dense NRSfM on the Grassmann manifold than [34]. Experimental results show that our method is as accurate as other algorithms and is numerically more efficient in handling noise. The main contributions of our work are as follows:

- An efficient framework for modeling non-rigidly deforming surface that exploits the advantage of Grassmann manifold representation of different dimensions based on its geometry.
- A formulation that encapsulates the local non-linearity of the deforming surface w.r.t its neighbors to enable the proper inference and representation of local linear subspaces.
- An iterative solution to the proposed cost function

based on ADMM [8], which is simple to implement and provide results as good as the best available methods. Additionally, it helps improve the 3D reconstruction substantially, in the case of noisy trajectories.

2. Related Work

A working solution to NRSfM was first introduced in the seminal work by Bregler *et al*. [9] which is an extension to Tomsai-Kannade rigid factorization method[48]. Although the problem still remains unsolved for arbitrary deformations, many profound works have been done to achieve a reliable solution to this problem under some or the other prior assumptions about the object [18, 3, 43, 40, 50, 37, 35, 23]. Since the literature on this topic is very extensive, we review the works that are of close relevance to the dense NRSfM methods under classical setting[2].

Earlier attempts to solve dense NRSfM used piecewise reconstruction of the shape parts which were further processed via a stitching step to get a global 3D shape [12, 45]. To our knowledge, Garg *et al*. variational approach [24] was the first to propose and demonstrate per pixel dense NRSfM algorithm without any 3D template prior. This method introduced a discrete total variational constraint with trace norm constraint on the global shape, which resulted in a biconvex optimization problem. Despite the algorithm outstanding results, it's computationally very expensive and needs a GPU to provide the solution.

In contrast, Dai *et al*. extended his simple prior free approach to solve dense NRSfM problem [18, 17]. The algorithm proposed a spatial-temporal formulation to solve the problem. The author revisits the temporal smoothness term from [18] and integrate it with a spatial smoothness term using the Laplacian of the non-rigid shape. The resultant optimization leads to a series of least squares to be minimized which makes it extremely slow to process. Recently, Kumar *et al*. modeled this problem on the Grassmann manifold [34]. Their work extended the spatiotemporal multibody framework to solve dense NRSfM [37]. The algorithm demonstrated that such an approach is more efficient, faster and accurate than all the other recent approaches to solve dense NRSfM task [24, 17, 4].

Consecutive frame-based approach has recently shown some promising results to solve dense 3D reconstruction of a general dynamic scene including non-rigid object [36, 44]. Nevertheless, motion segmentation, triangulation, as rigid as possible constraint and scale consistency quite often breaks down for the deforming object over frames. Therefore, dense NRSfM becomes extremely challenging for such algorithms. Not long ago, Gallardo *et al*. combined shading, motion and generic physical deformation to model dense NRSfM [21].

[2] By *classical setting*, we mean without using RGB-D or 3D template.

3. Preliminaries

In this paper, $\|.\|_{\mathtt{F}}$, $\|.\|_*$ denotes the Frobenius norm and nuclear norm respectively. $\|.\|_{\mathcal{G}}$ represent the notion of norm on the Grassmann manifold. Single angle bracket $< .,. >$ denotes the Euclidean inner product. For ease of understanding and completeness, in this section, we briefly review few important definitions related to the Grassmann manifold. Firstly, a manifold is a topological space that is locally similar to the Euclidean space. Out of several manifolds, the Grassmann manifold is a topologically rich non-linear manifold, each point of which represent the set of all right invariant subspace of the Euclidean space [19, 1, 34].

Definition 1. *The Grassmann manifold, denoted by $\mathcal{G}(\mathtt{p},\mathtt{d})$, consists of all the linear $\mathtt{p}$-dimensional subspace embedded in a '$\mathtt{d}$' dimensional Euclidean space $\mathbb{R}^{\mathtt{d}}$ such that $0 \leq \mathtt{p} \leq \mathtt{d}$ [Absil et al., 2009] [1].*

A point 'Φ' on the Grassmann manifold can be represented by $\mathbb{R}^{\mathtt{d}\times\mathtt{p}}$ matrix whose columns are composed of orthonormal basis. The space of such matrices with orthonormal columns is a Riemannian manifold such that $\Phi^{\mathtt{T}}\Phi = \mathbf{I}_{\mathtt{p}}$, where $\mathbf{I}_{\mathtt{p}}$ is a $\mathtt{p} \times \mathtt{p}$ identity matrix.

Definition 2. *Grassmann manifold can be embedded into the space of symmetric matrices via mapping $\Pi : \mathcal{G}(\mathtt{p},\mathtt{d}) \mapsto Sym(\mathtt{d}), \Pi(\Phi) = \Phi\Phi^{\mathtt{T}}$, where Φ is a Grassmann point [28, 30]. Given two Grassmann points Φ_1 and Φ_2, then the distance between them can be measured using the projection metric $d_g^2(\Phi_1,\Phi_2) = 0.5\|\Pi(\Phi_1) - \Pi(\Phi_2)\|_{\mathtt{F}}^2$ [28].*

In the past, these two properties of Grassmann manifold has been used in many computer vision applications [28, 10, 34]. Second definition is very important as it allows to measure the distance on the Grassmann manifold, hence, $(\mathcal{G}, d_g)$ forms a metric space. We used these properties in the construction of our formulation. For comprehensive details on this topic readers may refer to [28].

4. Problem Formulation

Let '$\mathtt{P}$' be the total number of feature points tracked across '$\mathtt{F}$' frames. Concatenating these 2D coordinates of each feature points for all frames across the columns of a matrix gives '$\mathtt{W}$' $\in \mathbb{R}^{2\mathtt{F}\times\mathtt{P}}$ matrix. This matrix is popularly known as *measurement matrix* [48]. Our goal is, given the image measurement matrix, estimate the camera motion and 3D coordinates of every 2D feature points across all frames.

We start our formulation with the classical representation to NRSfM *i.e.* $\mathtt{W} = \mathtt{RS}$, where, $\mathtt{R} \in \mathbb{R}^{2\mathtt{F}\times 3\mathtt{F}}$ is a block diagonal rotation matrix with each block as a 2×3 orthographic rotation matrix, and $\mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}}$ as the 3D structure matrix. With such a representation, the entire problem simplifies to the estimation of correct rotation matrix '$\mathtt{R}$' and

structure matrix 'S' such that the above relation holds. Following the assumption of the previous work [34], we estimate the rotation using Intersection method [18]. As a result, the task reduces to composing of an efficient algorithm that correctly models the surface deformations and provide better reconstruction results. Recent algorithms in NRSfM have demonstrated that clustering benefits reconstruction and vice-versa, however, the existing framework to employ this idea is not scalable to millions of points. To establish this idea for dense NRSfM, Kumar *et al.* [34] used LRR on Grassmannian manifold. Using the similar notions, we model dense deforming surface using Grassmannian representation to provide more reliable and accurate solution.

In the following subsection, we first introduce the Grassmannian representation of the surface and how to project these Grassmannians onto the lower dimension Grassmann manifold by preserving the neighboring information. In the later subsection, we use these representations to formulate the overall cost function for solving dense NRSfM problem.

4.1. Grassmannian representation

Let 'Φ_i' $\in \mathcal{G}(p, d)$ be a Grassmann point representing the ith local linear subspace spanned by i^{th} set of columns of 'S'. Using this notion, we decompose the entire trajectories of the structure into a set of 'K' Grassmannians $\xi = \{\Phi_1, \Phi_2, \Phi_3, ..., \Phi_K\}$. Now, such a representation treats each subspace independently and therefore, its low-dimensional linear representation may not be suitable to capture the surface dependent non-linearity. To properly represent Grassmannian which respects the neighboring non-linearity in low-dimension, we introduce a different strategy to model non-rigid surface in low-dimension. For now, let $\Delta \in \mathbb{R}^{d \times \tilde{d}}$ be a matrix that maps 'Φ_i' $\in \mathcal{G}(p, d)$ to 'ϕ_i' $\in \mathcal{G}(p, \tilde{d})$ such that $\tilde{d} < d$. Mathematically,

$$\phi_i = \Delta^T \Phi_i \tag{1}$$

Its quite easy to examine that ϕ_i is not a orthogonal matrix and, therefore, does not qualifies as a potential point on a Grassmann manifold. However, by performing a orthogonal-triangular (QR) decomposition of ϕ_i, we estimate the new representative of ϕ_i on the Grassmann manifold of '$\tilde{d}$' dimension.

$$\Theta_i U_i = \mathbf{qr}(\phi_i) = \Delta^T \Phi_i \tag{2}$$

Here, $\mathbf{qr}(.)$ is a function that returns the QR decomposition of the matrix. The $\Theta_i \in \mathbb{R}^{\tilde{d} \times p}$ is an orthogonal matrix and $U_i \in \mathbb{R}^{p \times p}$ is the upper triangular matrix[3]. Using Eq.(2), we represent the equivalence of Φ_i in low dimension as

$$\begin{aligned} \Theta_i &= \Delta^T (\Phi_i U_i^{-1}) \\ \Theta_i &= \Delta^T \Omega_i \end{aligned} \tag{3}$$

[3]Note: The value of $\tilde{d} \geq p$, Use $[\Theta_i, U_i] = \mathbf{qr}(\phi_i, 0)$ in MATLAB to get a square U_i matrix ($U_i \in \mathbb{R}^{p \times p}$)

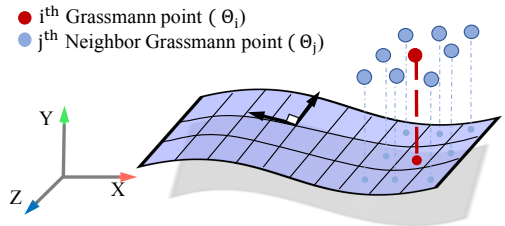


Figure 3: In contrast to [34], our modeling of surface using Grassmannians considers the similarity between the neighboring Grassmannians while representing it in the lower dimension. Based on the assumption that spatially neighboring surface tend to span similar subspace, defining neighboring subspace dependency graph is easy and most of the real-world examples follows such assumption. However, building such graph in shape space can be tricky.

where, $\Omega_i = \Phi_i U_i^{-1} \in \mathbb{R}^{d \times p}$. The key-point to note is that both Θ_i and ϕ_i has the same column space. In principle such a representation is useful however, it does not serve the purpose of preserving the non-linearity w.r.t its neighbors. In order to encapsulate the local dependencies (see Fig.(3), Fig.(4)), we further constrain our representation as:

$$E(\Delta) = \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Pi(\Theta_i) - \Pi(\Theta_j)\|_2^F \tag{4}$$

The parameter 'w_{ij}' accommodate the similarity knowledge between the two Grassmannians. Using the **Definition**(2) and Eq.(3), we further simplify Eq.(4) as

$$\begin{aligned} E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Delta^T \Omega_i \Omega_i^T \Delta - \Delta^T \Omega_j \Omega_j^T \Delta\|_F^2 \\ E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Delta^T (\Omega_i \Omega_i^T - \Omega_j \Omega_j^T) \Delta\|_F^2 \\ E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Delta^T (\Lambda_{ij}) \Delta\|_F^2 \end{aligned} \tag{5}$$

where, $\Lambda_{ij} \in Sym(d)$. The parameter 'w_{ij}' (similarity graph) is set as $exp(-d_g^2(\Phi_i, \Phi_j))$ with d_g as the projection metric (see **Definition** (2)). Eq.(5) is an unconstrained optimization problem and its solution may provide a trivial solution. To estimate the useful solution, we further constrain the problem. Using i^{th} Grassmann point 'Ω_i' and its neighbors, expand Eq.(5). By performing some simple algebraic manipulation, Eq.(5) reduces to

$$\mathbf{trace}\Big(\Delta^T \big(\sum_{i=1}^{K} \lambda_{ii} \Omega_i \Omega_i^T\big) \Delta\Big) \tag{6}$$

where, $\lambda_{ii} = \sum_{j=1}^{K} w_{ij}$. Constraining the value of Eq.(6)

to 1 provides the overall optimization for an efficient representation of the local non-rigid surface on the Grassmann manifold.

$$\mathtt{E}(\Delta) \equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathtt{i},\mathtt{j})}^{\mathtt{K}} \mathtt{w}_{\mathtt{ij}} \frac{1}{2} \|\Delta^{\mathtt{T}}(\Lambda_{\mathtt{ij}})\Delta\|_{\mathtt{F}}^2$$
$$\text{subject to:} \tag{7}$$
$$\mathbf{trace}\Big(\Delta^{\mathtt{T}}\big(\sum_{\mathtt{i}=1}^{\mathtt{K}} \lambda_{\mathtt{ii}}\Omega_{\mathtt{i}}\Omega_{\mathtt{i}}^{\mathtt{T}}\big)\Delta\Big) = 1$$

Its easy to verify that the matrix Λ and $\big(\sum_{\mathtt{i}=1}^{\mathtt{K}} \lambda_{\mathtt{ii}}\Omega_{\mathtt{i}}\Omega_{\mathtt{i}}^{\mathtt{T}}\big)$ are symmetric and positive semi-definite, and therefore, the above optimization can be solved as a generalized eigen value problem —refer supplementary material for details.

4.2. Dense NRSfM formulation

To solve the dense non-rigid structure from motion with the representation formulated in the previous sub-section §4.1, we propose to jointly optimize the objective function over the 3D structure and its local group representation. In order to build the overall objective function, we introduce each constraint equation one by one for clear understanding of our overall cost function.

$$\mathbf{E}_{\mathtt{p}}(\mathtt{S}) = \underset{\mathtt{S}}{\text{minimize}} \ \frac{1}{2}\|\mathtt{W} - \mathtt{RS}\|_{\mathtt{F}}^2 \tag{8}$$

The **first** term constrain the 3D structure such that it satisfies the re-projection error.

$$\mathbf{E}_{\mathtt{s}}(\mathtt{S}^{\sharp}) = \underset{\mathtt{S}^{\sharp}}{\text{minimize}} \ \|\mathtt{S}^{\sharp}\|_{*} \tag{9}$$

The **second** term caters the global assumption about the non-rigid object; that is the overall shape matrix is low-rank. To establish this assumption, we perform rank minimization of the shape matrix. Although the rank minimization of a matrix is NP-hard, it's relaxed to nuclear norm minimization to find an approximate solution. This term mainly penalizes the total number of independent shape required to represent the shape. The choice of minimizing $\mathtt{S}^{\sharp} \in \mathbb{R}^{3\mathtt{P}\times\mathtt{F}}$ instead to $\mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}}$ is inspired from Dai *et al.*'s work [18]. Since the dense deforming shape is composed of several local linear low-dimensional subspace, the global constraint (Eq.(9)) may not reflect their local dependency. Therefore, in order to introduce the local subspace constraint on the shape, we use the notion of self-expressiveness on the non-linear Grassmann manifold space.

$$\underset{\mathtt{E},\mathtt{C},\mathtt{S}^{\sharp}}{\text{minimize}} \ \|\mathtt{E}\|_{\mathcal{G}}^2 + \beta_2\|\mathtt{S}^{\sharp}\|_{*} + \beta_3\|\mathtt{C}\|_{*}$$
$$\text{subject to: } \mathtt{S}^{\sharp} = f(\mathtt{S}), \mathtt{S} = \mathtt{SC} + \mathtt{E} \tag{10}$$

Here, we define $f : \mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}} \mapsto \mathtt{S}^{\sharp} \in \mathbb{R}^{3\mathtt{P}\times\mathtt{F}}$ and $\mathtt{C} \in \mathbb{R}^{\mathtt{P}\times\mathtt{P}}$ as the coefficient matrix. We know from the literature that the Grassmann manifold is isometrically equivalent to the symmetric idempotent matrix [11]. So, we embed

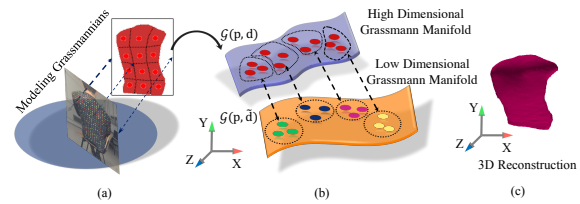


Figure 4: Conceptual illustration of our modeling (a) Modeling of 3D trajectories to Grassmann points (b) The two grassmann manifold and mapping of the points between them to infer better cluster index that leads to better reconstruction (c) The 3D reconstruction of the non-rigid deforming object.

the Grassmann manifold into symmetric matrix manifold to define the self-expressiveness. Let $\tilde{\xi} = \{\Theta_1, \Theta_2, ..., \Theta_{\mathtt{K}}\}$ be the set of Grassmannians on a low-dimensional Grassmann manifold. The elements of $\tilde{\xi}$ are the projection of high-dimensional Grassmannian representation of the columns of 'S' matrix. Let $\chi = \{(\Theta_1\Theta_1^{\mathtt{T}}), (\Theta_2\Theta_2^{\mathtt{T}}), ..., (\Theta_{\mathtt{K}}\Theta_{\mathtt{K}}^{\mathtt{T}})\}$ be its embedding onto symmetric matrix manifold. Using such embedding techniques we re-write Eq.(10) as

$$\underset{\mathtt{E},\tilde{\mathtt{C}},\mathtt{S}^{\sharp}}{\text{minimize}} \ \|\mathtt{E}\|_{\mathtt{F}}^2 + \beta_2\|\mathtt{S}^{\sharp}\|_{*} + \beta_3\|\tilde{\mathtt{C}}\|_{*}$$
$$\text{subject to:} \tag{11}$$
$$\mathtt{S}^{\sharp} = f(\mathtt{S}), \chi = \chi\tilde{\mathtt{C}} + \mathtt{E}$$

where, $\tilde{\mathtt{C}} \in \mathbb{R}^{\mathtt{K}\times\mathtt{K}}$ and $\chi \in \mathbb{R}^{\tilde{\mathtt{d}}\times\tilde{\mathtt{d}}\times\mathtt{K}}$ denotes the coefficient matrix of Grassmannians and structure tensor respectively, with $\mathtt{K}$ as the total number of Grassmannians. Generally, $\mathtt{K} << \mathtt{P}$, which makes such representation scalable.

The **third** term we introduce is composed of few constraint functions that provides a way to group Grassmannians and recover 3D shape simultaneously. Let $\mathbf{P} \in \mathbb{R}^{1\times\mathtt{P}}$ be an ordering vector that contains the index of columns of $\mathtt{S}$. Our function definition is of the form $\{(\text{output}, \text{function}(.)) : \text{definition}\}$. Using it, we define the function f_g, f_h, f_p and f_s as follows:

$$\big\{\big(\xi, f_g(\mathbf{P}, \mathtt{S})\big) : \text{order } \{\mathtt{S}_{\mathtt{i}}\}_{\mathtt{i}=1}^{\mathtt{K}} \text{ columns of } \mathtt{S} \text{ of using } \mathbf{P},$$
$$\xi := \{\Phi_{\mathtt{i}}\}_{\mathtt{i}=1}^{\mathtt{K}} \text{where}, [\,\Phi_{\mathtt{i}}, \Sigma_{\mathtt{i}}, \xi_{\mathtt{vi}}] = \text{svds}(\mathtt{S}_{\mathtt{i}}, \mathtt{p})\big\} \tag{12}$$

$$\big\{\big(\tilde{\xi}, f_h(\Delta, \xi)\big) : \tilde{\xi} = \{\Theta_{\mathtt{i}}\}_{\mathtt{i}=1}^{\mathtt{K}}, \Theta_{\mathtt{i}} = \Delta^{\mathtt{T}}(\Phi_{\mathtt{i}}\mathtt{U}_{\mathtt{i}}^{-1}),$$
$$\text{where, } \Delta = \text{solution to the minimization of Eq.(7)}\big\} \tag{13}$$

$$\big\{(\mathbf{P}, f_p(\tilde{\xi}, \tilde{\mathtt{C}}, \mathbf{P}_{\mathtt{o}}) : \mathbf{P} = \text{spectral_clustering}(\tilde{\xi}, \tilde{\mathtt{C}}, \mathbf{P}_{\mathtt{o}})\big\} \tag{14}$$

$$\big\{\big(\mathtt{S}, f_s(\xi, \Sigma, \xi_{\mathtt{v}})\big) : \mathtt{S}_{\mathtt{i}} = [\xi_{\mathtt{i}}\ \Sigma_{\mathtt{i}}\ \xi_{\mathtt{vi}}], \text{where } \Sigma_{\mathtt{i}} \in \mathbb{R}^{\mathtt{P}\times\mathtt{P}}\big\} \tag{15}$$

Intuitively, the first function (f_g) uses the ordering vector $\mathbf{P} \in \mathbb{R}^{1\times\mathtt{P}}$ to refine the grouping of the trajectories for suitable Grassmannian representation. The second function

(f_h) projects the Grassmannians to a lower dimension in accordance with the neighbors using Eq.(7). The third function (f_p) uses the projected Grassmannians to assign proper labeling to the Grassmann points and update the given ordering vector $\mathbf{P}$ using spectral clustering. The fourth function (f_s) uses the group of trajectories to reconstruct back the set of local surface. Σ, $\xi_{\mathtt{v}}$ are the singular values and right singular vector matrices in the high-dimension.

Objective Function: Combining all the above terms and constraints provides our overall cost function.

$$\underset{\mathtt{E},\tilde{\mathtt{C}},\mathtt{S},\mathtt{S}^\sharp}{\text{minimize}} \frac{1}{2}\|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2+\beta_1\|\mathtt{E}\|_{\mathtt{F}}^2+\beta_2\|\mathtt{S}^\sharp\|_*+\beta_3\|\tilde{\mathtt{C}}\|_*$$

subject to:

$$\mathtt{S}^\sharp = f(\mathtt{S}), \chi = \chi\tilde{\mathtt{C}}+\mathtt{E},$$
$$\xi = f_g(\mathbf{P},\mathtt{S}), \tilde{\xi} = f_h(\Delta,\xi),$$
$$\mathtt{S} = f_s(\xi,\Sigma,\xi_{\mathtt{v}}), \mathbf{P} = f_p(\tilde{\xi},\tilde{\mathtt{C}},\mathbf{P}_\circ) \tag{16}$$

where $\mathbf{P}_\circ$ vector contains the initial ordering of the columns of 'W' and 'S'. The function (f_p) provides the ordering index to rearrange the columns of 'S' matrix to be consistent with 'W' matrix. This is important because, grouping the set of columns of 'S' over iteration, disturbs its initial arrangements.

5. Solution

The optimization proposed in Eq.(16) is a coupled optimization problem. Several methods of Bi-level optimization can be used to solve such minimization problem [6, 27]. Nevertheless, we propose ADMM [8] based solution due to its application in many non-convex optimization problems. The key point to note is that one of our constraint is composed of separate optimization problem (f_h) *i.e.*, the solution to Eq.(7), and therefore, we cannot directly embed the constraint to the main objective function. Instead, we only introduce two Lagrange multiplier $\mathbf{L_1}, \mathbf{L_2}$ to concatenate a couple of constraints back to the original objective function. The remaining constraints are enforced over iteration. To decouple the variable $\tilde{\mathtt{C}}$ from χ, we introduce auxiliary variable $\tilde{\mathtt{C}} = \mathtt{Z}$. We apply these operations to our optimization problem to get the following Augmented Lagrangian form:

$$\underset{\mathtt{Z},\tilde{\mathtt{C}},\mathtt{S},\mathtt{S}^\sharp}{\text{minimize}} \frac{1}{2}\|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2+\beta_1\|\chi-\chi\tilde{\mathtt{C}}\|_{\mathtt{F}}^2+\beta_2\|\mathtt{S}^\sharp\|_*+$$
$$\frac{\rho}{2}\|\mathtt{S}^\sharp-f(\mathtt{S})\|_{\mathtt{F}}^2+<\mathbf{L_1},\mathtt{S}^\sharp-f(\mathtt{S})>+\beta_3\|\mathtt{Z}\|_*+$$
$$\frac{\rho}{2}\|\tilde{\mathtt{C}}-\mathtt{Z}\|_{\mathtt{F}}^2+<\mathbf{L_2},\tilde{\mathtt{C}}-\mathtt{Z}>$$

subject to: $\xi = f_g(\mathbf{P},\mathtt{S}), \tilde{\xi} = f_h(\Delta,\xi),$

$$\mathtt{S} = f_s(\xi,\Sigma,\xi_{\mathtt{v}}), \mathbf{P} = f_p(\tilde{\xi},\tilde{\mathtt{C}},\mathbf{P}_\circ) \tag{17}$$

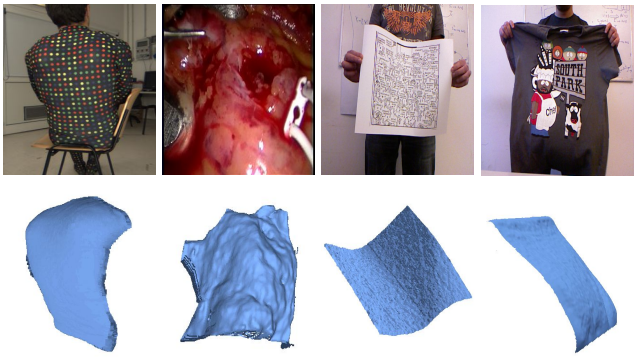
Algorithm 1 Dense Non-rigid Structure from Motion (**MoJu**)

Require: $\mathtt{W}$, $\mathtt{R}$, $\{\beta_{\mathtt{i}}\}_{\mathtt{i}=1}^3$, ρ=e^{-2}, ρ_m=e^8, ϵ=e^{-10}, c =1.1, $\mathtt{K}$;
Initialize: $\mathtt{S}$=**pinv**($\mathtt{R}$)$\mathtt{W}$, $\mathtt{S}^\sharp$=$f(\mathtt{S})$, $\mathtt{Z}$=**0**, $\{\mathbf{L}_{\mathtt{i}}\}_{\mathtt{i}=1}^2$=**0**, $\tilde{\mathtt{d}}$;
Δ = [$\mathbf{I}_{\tilde{\mathtt{d}}\times\tilde{\mathtt{d}}}$; random values], p %top singular values
$\mathbf{P}_\circ$ = kmeans++($\mathtt{S}$, $\mathtt{K}$), iter = 1, $\mathbf{P}_{\text{store}}$(iter, :) = $\mathbf{P}_\circ$,
$\mathbf{P}$ = $\mathbf{P}_\circ$
while not converged **do**
1. $\mathtt{S} := \text{mldivide}\big(\mathtt{R}^{\mathtt{T}}\mathtt{R}+\rho\mathtt{I}, \rho(f^{-1}(\mathtt{S}^\sharp)+\frac{f^{-1}(\mathbf{L_1})}{\rho})+\mathtt{R}^{\mathtt{T}}\mathtt{W}\big)$;
2. $\xi := f_g(\mathbf{P},\mathtt{S})$; see Eq.(12)
3. $\mathtt{W} := \text{arrange_column}(\mathbf{P},\mathtt{W})$
4. Update the similarity matrix '$\mathtt{w}_{\mathtt{ij}}$' using ξ. §4.1
5. $\tilde{\xi} := f_h(\xi,\Delta)$; s.t, $\Delta \equiv \underset{\Delta}{\text{minimize}}\ \mathtt{E}(\Delta)$; see Eq.(13)
6. $\Gamma_{\mathtt{ij}} = \text{Tr}[(\Theta_{\mathtt{j}}^{\mathtt{T}}\Theta_{\mathtt{i}})((\Theta_{\mathtt{i}}^{\mathtt{T}}\Theta_{\mathtt{j}})]; \Gamma = (\Gamma_{\mathtt{ij}})_{\mathtt{ij}=1}^{\mathtt{K}}; \mathtt{L} = \text{chol}(\Gamma)$
7. $\tilde{\mathtt{C}} := \Big(2\beta_1\mathtt{LL}^{\mathtt{T}}+\rho(\mathtt{Z}-\frac{\mathbf{L_2}}{\rho})\Big)\ \big(2\beta_1\mathtt{LL}^{\mathtt{T}}+\rho\mathtt{I}\big)^{-1}$;
8. $\mathbf{P} := f_p(\tilde{\xi},\tilde{\mathtt{C}},\mathbf{P})$;
9. $\mathtt{S} := f_s(\xi,\Sigma,\xi_v)$; see Eq.(14)
10. $\mathtt{S}^\sharp := \mathtt{U_s}\mathcal{S}_{\frac{\beta_2}{\rho}}(\Sigma_{\mathtt{s}})\mathtt{V_s}$; s.t, $[\mathtt{U_s},\Sigma_{\mathtt{s}},\mathtt{V_s}] := \text{svd}(f(\mathtt{S})-\frac{\mathbf{L_1}}{\rho})$
11. $\mathtt{Z} := \mathtt{U_z}\mathcal{S}_{\frac{\beta_3}{\rho}}(\Sigma_{\mathtt{z}})\mathtt{V_z}$; s.t, $[\mathtt{U_z},\Sigma_{\mathtt{z}},\mathtt{V_z}] := \text{svd}(\tilde{\mathtt{C}}+\frac{\mathbf{L_2}}{\rho})$;
12. $\mathbf{L_1} := \mathbf{L_1}+\rho(\mathtt{S}^\sharp-f(\mathtt{S}))$; $\mathbf{L_2} := \mathbf{L_2}+\rho(\tilde{\mathtt{C}}-\mathtt{Z})$
13. iter := iter + 1; $\mathbf{P}_{\text{store}}$(iter, :) := $\mathbf{P}$;
14. $\rho := \min(\rho_m, c\rho)$;
15. gap := $\max\{\|\mathtt{S}^\sharp-f(\mathtt{S})\|_\infty, \|\tilde{\mathtt{C}}-\mathtt{Z}\|_\infty\}$;
$(\text{gap}<\epsilon)\vee(\rho>\rho_m)\rightarrow$ **break**; %convergence check
end while
return $\mathtt{S}$;
$e_{3\mathtt{D}}$ = **Estimate_error** ($\mathtt{S}, \mathtt{S}_{\mathtt{GT}}, \mathbf{P}_{\text{store}}$); %use Eq.(18)

Note that $\tilde{\mathtt{C}}$ provides the information about the subspace, not the vectorial points. However, we have the chart of the trajectories and its corresponding subspace. Once, we group the trajectories based on $\tilde{\mathtt{C}}$, $f_g(.)$ provides new Grassmann sample corresponding to each group. The definition of $f_h(.)$ and $f_s(.)$ is provided in Eq.(7) and Eq.(14) respectively. More generally, the solution to the optimization in Eq.(7) is obtained by solving it as a generalized eigenvalue problem. To keep the order of columns of 'S' matrix consistent with 'W' matrix $f_p(.)$ provides the ordering index. We provide the implementation details of our method with suitable MATLAB commands in the Algorithm Table (1). For details on the derivation to each sub-problem, kindly refer to the supplementary material.

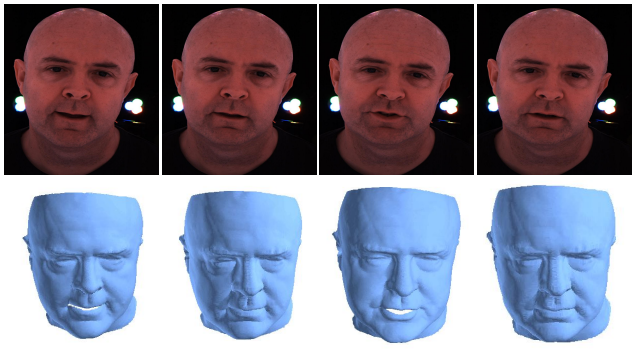
6. Initialization and Evaluation

We performed experiments and evaluation on the available standard benchmark datasets [23, 49, 7]. To keep our evaluations consistent with the previous methods, we compute the mean normalized 3D reconstruction error of the



(a) (b) (c) (d)

Figure 5: From left: 3D reconstruction results on Back [23], Heart [23], Paper[49] and T-shirt [49] data sequence respectively.



(a) (b) (c) (d)

Figure 6: 3D reconstruction results on the Actor sequence [7].

estimated shape '$\mathtt{S}_{\text{est}}$' after convergence as

$$\mathtt{e}_{3D} = \frac{1}{\mathtt{F}} \sum_{\mathtt{i}=1}^{\mathtt{F}} \frac{\|\mathtt{S}^{\mathtt{i}}_{\text{est}} - \mathtt{S}^{\mathtt{i}}_{\mathtt{GT}}\|_{\mathtt{F}}}{\|\mathtt{S}^{\mathtt{i}}_{\mathtt{GT}}\|_{\mathtt{F}}} \tag{18}$$

here '$\mathtt{S}_{\mathtt{GT}}$' denotes the ground-truth 3D shape matrix.

Initialization: We used Intersection method [18] to estimate the rotation matrix and initialize $\mathtt{S} = \textbf{pinv}(\mathtt{R})\mathtt{W}$. The initial grouping of the trajectories or columns of $\mathtt{S}$ is done using k-means++ algorithm [5]. These groups are then used to initialize $\mathbf{P}_{\circ}$, $\mathbf{P}$ and the Grassmann points $\{\Phi_{\mathtt{i}}\}_{\mathtt{i}=1}^{\mathtt{K}} \in \xi$ via subset of singular vectors. To represent the Grassmannians in the lower-dimension, we solve Eq.(7) to initialize $\tilde{\xi}$ and store corresponding singular values. The similarity matrix or graph in Eq.(7) is build using the distance measure between the Grassmannians in the embedding space §4.1.

1. Results on synthetic Face dataset: The synthetic face dataset is composed of four distinct sequence [23] with 28,880 feature points tracked over multiple frames. Each sequence captures the human facial expression with a different range of deformations and camera motion. Sequence 1 and Sequence 2 are 10 frame long video with rotation in the range $\pm 30°$ and $\pm 90°$ respectively. Sequence 3 and Sequence 4 are 99 frame long video that contains high frequencies and low frequencies rotation respectively which captures real human facial deformations. Table (1) shows the statistical results obtained on these sequences using our algorithm. For qualitative results on these sequences kindly refer to the supplementary material.

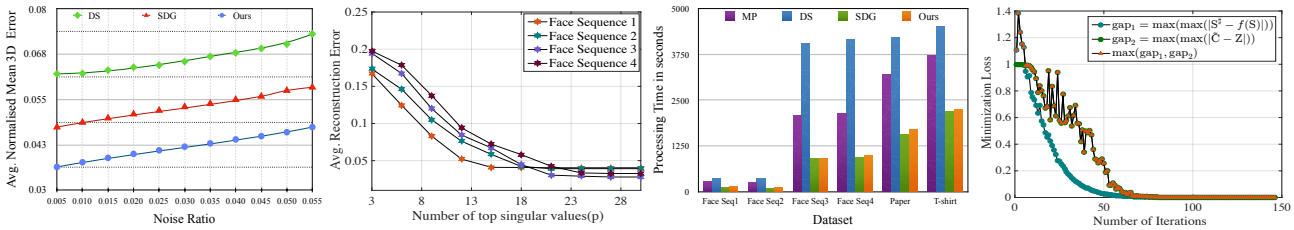
2. Results on Paper and T-shirt dataset: Varol *et al*. introduced 'kinect_paper' and 'kinect_tshirt' datasets to test the performance of NRSfM algorithm under real conditions [49]. This dataset provides sparse SIFT [41] feature tracks and noisy depth information captured from Microsoft Kinect for all the frames. As a result, to get dense 2D feature correspondences of the non-rigid object for all the frames becomes difficult. To circumvent this issue, we used Garg *et al*. [22] algorithm to estimate the measurement matrix. To keep the numerical comparison consistent with the previous work in dense NRSfM [34], we used the same coordinate range for tracking the features. Numerically, its $\mathtt{x}_{\mathtt{w}} = (253, 253, 508, 508)$, $\mathtt{y}_{\mathtt{w}} = (132, 363, 363, 132)$ rectangular window across 193 frames for kinect_paper sequence. For kinect_tshirt sequence, we considered rectangular window of $\mathtt{x}_{\mathtt{w}} = (203, 203, 468, 468)$, $\mathtt{y}_{\mathtt{w}} = (112, 403, 403, 112)$ across 313 frames, same as used in Kumar *et al*. work [34]. Fig.(5) shows the reconstruction results on these sequence with comparative results provided in Table (1).

3. Results on Actor dataset: Beeler *et al*. [7] introduced Actor dataset for high-quality facial performance capture. This dataset is composed of 346 frames captured from seven cameras with 1,180,232 vertices. The dataset captures the fine details of facial expressions which is extremely useful in the testing of NRSfM algorithms. Nevertheless, for our experiment, we require dense 2D image feature correspondences across all images as input, which we synthesized using ground-truth 3D points and synthetically generated orthographic camera rotations. To maintain the consistency with the previous works in dense NRSfM for performance evaluations, we synthesized two different datasets namely Actor Sequence1 and Actor Sequence2 based on the head movement as described in Ansari *et al*. work [4]. Fig.(6) shows the dense detailed reconstruction that is achieved using our algorithm. Table (1) clearly indicates the benefit of our approach to reconstruct such complex deformations.

4. Results on Face, Heart, Back dataset: To evaluate the variational approach to dense NRSfM [23] Garg *et al*. introduced these datasets. Its sequences are composed of monocular video's captured in a natural environment with varying lighting condition and large displacements. It consists of three different videos with 120, 150 and 80 frames for face sequence, back sequence and heart sequence respectively. Additionally, it provides dense 2D feature track for the same with 28332, 20561, and 68295 features track over

Dataset ↓ / Method →	MP [42]	PTA [2]	CSF1 [25]	CSF2[26]	DV [23]	DS [17]	SMSR [4]	SDG[34]	Ours
Face Sequence 1	0.2572	0.1559	0.5325	0.4677	0.0531	0.0636	0.1893	0.0443	**0.0404**
Face Sequence 2	0.0644	0.1503	0.9266	0.7909	0.0457	0.0569	0.2133	**0.0381**	0.0392
Face Sequence 3	0.0682	0.1252	0.5274	0.5474	0.0346	0.0374	0.1345	0.0294	**0.0280**
Face Sequence 4	0.0762	0.1348	0.5392	0.5292	0.0379	0.0428	0.0984	**0.0309**	0.0327
Actor Sequence 1	0.5226	0.0418	0.3711	0.3708	-	0.0891	0.0352	0.0340	**0.0274**
Actor Sequence 2	0.2737	0.0532	0.2275	0.2279	-	0.0822	0.0334	0.0342	**0.0289**
Paper Sequence	0.0827	0.0918	0.0842	0.0801	-	0.0612	-	0.0394	**0.0338**
T-shirt Sequence	0.0741	0.0712	0.0644	0.0628	-	0.0636	-	**0.0362**	0.0386

Table 1: Statistical comparison of our method with other competing approaches. Quantitative evaluations for SMSR [4] and DV [23] are not performed by us due to the unavailability of their code, and therefore, we tabulated their reconstruction error from their published work.



(a) (b) (c) (d)

Figure 7: (a) Variation in the average 3D reconstruction error with change in the noise ratio's for face dataset[23]. (b) Fluctuation in the 3D reconstruction accuracy with change number of top singular values and corresponding singular vectors used by our algorithm for face sequence[23]. (c) Processing time againt other competing algorithm's on Intel Core i7-4790 CPU @3.60GHz x 8 Desktop with MATLAB 2017b, our method show comparable execution timing to SDG[34]. (d) A typical ADMM optimization convergence curve of our algorithm.

the frames for face, back and heart sequence. No ground-truth 3D is available with this dataset for evaluation. Fig.(5) show reconstruction results on back and heart sequence. For more qualitative results on these sequences, kindly refer to the supplementary material.

6.1. Algorithmic Analysis

We performed some more experiments to understand the behavior of our algorithm under different input parameters and evaluation setups. In practice these experiment help analyze the practical applicability of our algorithm.

1. Performance over noisy trajectories: We utilized the standard experimental procedure to analyze the behavior of our algorithm under different noise levels. Similar to the work of Kumar *et al*. [34], we added the Gaussian noise to the input trajectories. The standard deviation of the noise are adjusted as $\sigma_g = \lambda_g \max\{|\mathtt{W}|\}$ with λ_g varying from 0.01 to 0.055. Fig.(7(a)) show the quantitative comparison of our approach with recent algorithms namely DS [17] and SDG [34]. The graph is plotted by taking the average reconstruction error of all the four synthetic face dataset [23]. The procured statistics indicate that our algorithm is more resilient to noise than other competing methods.

2. Performance with change in the number of singular values: The selection of 'p' in $\mathcal{G}(\mathtt{p},\mathtt{d})$ *i.e.* the number of top singular vectors for Grassmannian representation and its corresponding singular values to perform reconstruction can directly affect the performance of our algorithm. However, it has been observed over several experiments that we need very few singular value and singular vectors to recover dense detailed 3D reconstruction of the deforming object. Fig.(7(b)) show the variation in average 3D reconstruction with the values of 'p' for synthetic face dataset [23].

3. Processing Time and Convergence: Our algorithm execution time is almost at par or a bit slower than SDG [34]. Fig.(7(c)) show the processing time taken by our method on different datasets. Fig.(7(d)) show a typical convergence curve of our algorithm. Ideally, it takes 120-150 iteration to provide an optimal solution to the problem.

7. Conclusion

Our Grassmannian representation of a non-rigidly deforming surface exploits the advantage of Grassmannians of different dimensions to jointly estimate better grouping of subspaces and their corresponding 3D geometry. Our approach explicitly leverages the geometric structure of the non-rigidly moving object w.r.t its neighbors on manifold via similarity graph and, it's embedding in the lower dimension. We empirically demonstrated that our method is able to achieve 3D reconstruction accuracy which is better or as good as the state-of-the-art, with significant improvement in handling noisy trajectories.

Acknowledgement. The author was supported in part by the ARC Centre of Excellence for Robotic Vision (CE140100016), ARC Discovery

project on 3D computer vision for geo-spatial localisation (DP190102261) and ARC DECRA project (DE140100180). The author thank his elder brother Aditya Kumar for his relentless support and advise. Thank you Roya Safaei for her constant help and constructive suggestions. Also, the author thank Dr. J. Varela and Dr. C. Punch, ACT for the medical treatment during the course of this work.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *Advances in neural information processing systems*, pages 41–48, 2009.

[3] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, 2011.

[4] Mohammad Dawud Ansari, Vladislav Golyanik, and Didier Stricker. Scalable dense monocular surface reconstruction. *arXiv preprint arXiv:1710.06130*, 2017.

[5] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[6] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

[7] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. In *ACM Transactions on Graphics (TOG)*, volume 30, page 75. ACM, 2011.

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[9] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 690–696. IEEE, 2000.

[10] Hasan Ertan Cetingul and René Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1896–1902, 2009.

[11] Yasuko Chikuse. *Statistics on special manifolds*, volume 174. Springer Science & Business Media, 2012.

[12] T Collins and A Bartoli. Locally affine and planar deformable surface reconstruction from video. In *International Workshop on Vision, Modeling and Visualization*, pages 339–346, 2010.

[13] Keenan Crane. *Conformal geometry processing*. California Institute of Technology, 2013.

[14] Keenan Crane. Discrete differential geometry: An applied introduction. 2015.

[15] Keenan Crane, Fernando De Goes, Mathieu Desbrun, and Peter Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*, page 7. ACM, 2013.

[16] Keenan Crane and Max Wardetzky. A glimpse into discrete differential geometry. *Notices of the American Mathematical Society*, 64(10):1153–1159, November 2017.

[17] Yuchao Dai, Huizhong Deng, and Mingyi He. Dense non-rigid structure-from-motion made easy-a spatial-temporal smoothness based solution. *arXiv preprint arXiv:1706.08629*, 2017.

[18] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014.

[19] Piotr Dollár, Vincent Rabaud, and Serge Belongie. Non-isometric manifold learning: Analysis and an algorithm. In *International Conference on Machine Learning*, pages 241–248, 2007.

[20] Marián Fecko. *Differential geometry and Lie groups for physicists*. Cambridge University Press, 2006.

[21] Mathias Gallardo, Toby Collins, Adrien Bartoli, and France Mathias. Dense non-rigid structure-from-motion and shading with unknown albedos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3884–3892, 2017.

[22] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Robust trajectory-space tv-l1 optical flow for non-rigid sequences. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 300–314. Springer, 2011.

[23] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1272–1279, 2013.

[24] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. A variational approach to video registration with subspace constraints. *International journal of computer vision*, 104(3):286–314, 2013.

[25] Paulo FU Gotardo and Aleix M Martinez. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2051–2065, 2011.

[26] Paulo FU Gotardo and Aleix M Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3065–3072. IEEE, 2011.

[27] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[28] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *International conference on Machine learning*, pages 376–383. ACM, 2008.

[29] Onur C Hamsici, Paulo FU Gotardo, and Aleix M Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *European Conference on Computer Vision*, pages 260–273. Springer, 2012.

[30] Mehrtash Harandi, Conrad Sanderson, Chunhua Shen, and Brian Lovell. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3120–3127. IEEE, 2013.

[31] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *European conference on computer vision*, pages 17–32. Springer, 2014.

[32] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 140–149, 2015.

[33] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. *arXiv preprint arXiv:1902.10274*, 2019.

[34] Suryansh Kumar, Anoop Cherian, Yuchao Dai, and Hongdong Li. Scalable dense non-rigid structure-from-motion: A grassmannian perspective. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[35] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 148–156. IEEE, 2016.

[36] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4649–4657, Oct 2017.

[37] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion. *Pattern Recognition*, 71:428–443, May 2017.

[38] Suryansh Kumar, Ram Srivatsav Ghorakavi, Yuchao Dai, and Hongdong Li. Dense depth estimation of a complex dynamic scene without explicit 3d motion estimation. *arXiv preprint arXiv:1902.03791*, 2019.

[39] Viktor Larsson and Carl Olsson. Compact matrix factorization with dependent subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017*, volume 2017, pages 4361–4370. Institute of Electrical and Electronics Engineers Inc., 2017.

[40] Minsik Lee, Jungchan Cho, Chong-Ho Choi, and Songhwai Oh. Procrustean normal distribution for non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1280–1287, 2013.

[41] David G Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer vision*, pages 1150–1157, 1999.

[42] Marco Paladini, Alessio Del Bue, Joao Xavier, Lourdes Agapito, Marko Stovsic, and Marija Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *International journal of computer vision*, 96(2):252–276, 2012.

[43] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *European conference on computer vision*, pages 158–171. Springer, 2010.

[44] René Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4058–4066, 2016.

[45] Chris Russell, Joao Fayad, and Lourdes Agapito. Dense non-rigid structure from motion. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 509–516, 2012.

[46] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[47] Jonathan Taylor, Allan D Jepson, and Kiriakos N Kutulakos. Non-rigid structure from locally-rigid motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2761–2768. IEEE, 2010.

[48] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[49] Aydin Varol, Mathieu Salzmann, Pascal Fua, and Raquel Urtasun. A constrained latent variable model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2248–2255. Ieee, 2012.

[50] Yingying Zhu, Dong Huang, Fernando De La Torre, and Simon Lucey. Complex non-rigid motion 3d reconstruction by union of subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1542–1549, 2014.