Dense Non-Rigid Structure from Motion: A Manifold Viewpoint

Suryansh Kumar, Luc Van Gool, Carlos E. P. de Oliveira, Anoop Cherian, Yuchao Dai, Hongdong Li

Abstract—Non-Rigid Structure-from-Motion (NRSfM) problem aims to recover 3D geometry of a deforming object from its 2D feature correspondences across multiple frames. Classical approaches to this problem assume a small number of feature points and, ignore the local non-linearities of the shape deformation, and therefore, struggles to reliably model non-linear deformations. Furthermore, available dense NRSfM algorithms are often hurdled by scalability, computations, noisy measurements and, restricted to model just global deformation. In this paper, we propose algorithms that can overcome these limitations with the previous methods and, at the same time, can recover a reliable dense 3D structure of a non-rigid object with higher accuracy. Assuming that a deforming shape is composed of a union of *local* linear subspace and, span a *global* low-rank space over multiple frames enables us to efficiently model complex non-rigid deformations. To that end, each local linear subspace is represented using Grassmannians and, the global 3D shape across multiple frames is represented using a low-rank representation. We show that our approach significantly improves accuracy, scalability, and robustness against noise. Also, our representation naturally allows for simultaneous reconstruction and clustering framework which in general is observed to be more suitable for NRSfM problems. Our method currently achieves leading performance on the standard benchmark datasets.

Index Terms—Non-Rigid Structure from Motion, Linear Subspace, Low-Rank, Grassmann Manifold.

1 Introduction

In this work, we will be focused on the problem of Dense Non-rigid Structure from Motion (NRSfM). Generally, the goal of this problem is to solve dense 3D shape of a non-rigidly deforming object in the scene from its per pixel image correspondences across multiple frames. Application that benefits from dense NRSfM includes animation [1], motion capture [2], 3D facial expression capture [3], human heart 3D model for bypass surgery [3] and many more. These examples demonstrate that NRSfM is central to a wide range of important real-world applications and therefore, a reliable solution to NRSfM can benefit several areas in science and engineering.

There are different ways to solve non-rigid structure-from-motion problem, among them, matrix factorization is one of the most popular and a well-known approach to find a solution to this problem [4] [2] [5] [6]. Under the matrix factorization approach, a measurement matrix (a matrix with 2D trajectories as its column vectors) is decomposed into a motion matrix and a shape matrix §4.1. Consequently, any solution to this problem using this approach depends on the proper modeling of *structure*, and an efficient approach to estimate *motion*. Mathematically, one can assume that the 3D shape belongs to some *shape* manifold[1] and the motion lies on a differentiable manifold [7]. Keeping this perspective to solve dense NRSfM, it's quite natural to think of this problem in terms of manifolds, and how to model this problem efficiently using manifold representation.

Our survey reveals that the advancements in the non-rigid structure-from-motion for *sparse* set of points has been steady over the years [2] [6] [8] [9] [10] [5] [11] [12] [13] [14] [15], yet, the developments in dense NRSfM algorithm has been limited [3] [16] [17]. The reason for such a limited development in dense NRSfM is perhaps due to its dependence on per pixel reliable 2D image correspondences, across multiple frames, or the absence of resilient mathematical representation to model dense surface deformation. One can argue on the efficient motion estimation, however, from image correspondences, we can only estimate relative motion, and reliable algorithms with convincing theory exists to perform this task well [5] [11]. Additionally, with the recent developments in learning based approaches, per pixel correspondences can be achieved with a remarkable accuracy [18] [19], which leaves dense non-rigid shape representation and its modeling as a potential gray area for research in **dense** NRSfM.

A natural way to deal with dense NRSfM is to try classical sparse NRSfM algorithms, which in fact, works quite well for a few sets of points. Our experiments show that the existing sparse NRSfM algorithms do not cascade well to dense NRSfM settings. This is because the assumption and the formulation developed for the sparse NRSfM does not hold entirely for dense deforming surfaces. For example: The assumption that a non-rigid shape spans a global low-rank space [5] [11]. Now, such an assumption may hold for the global structure of the problem, however, it fails to cater the inherent local deformation of the shape over time and space. Therefore, dense NRSfM solution using sparse NRSfM formulations provides implausible results. This drawback with [5] [11] led to the development of union of subspace based methods in NRSfM [20] [13] [21]. Among these methods, Kumar et.al. work on the union of subspace demonstrated state-of-the-art results [13] in the NRSfM challenge at CVPR 2017 [22]. Nevertheless, these algorithms do not scale to dense feature points and their resilience

- *Suryansh Kumar, Luc Van Gool, Carlos E.P Oliveira is with Computer Vision Lab at ETH Zürich, Switzerland.*
  E-mail: {sukumar, vangool, coliveira}@vision.ee.ethz.ch
- *Anoop Cherian is with MERL Cambridge, MA, USA.*
  E-mail: cherian@merl.com
- *Yuchao Dai is with Northwestern Polytechnical University, X'ian China*
  E-mail: daiyuchao@gmail.com
- *Hongdong Li is with Australian National University, Canberra*
  E-mail: hongdong.li@anu.edu.au

[1] Here, we assume a smooth, continuous surface for dense NRSfM problem

to noise and outliers remains unsatisfactory.

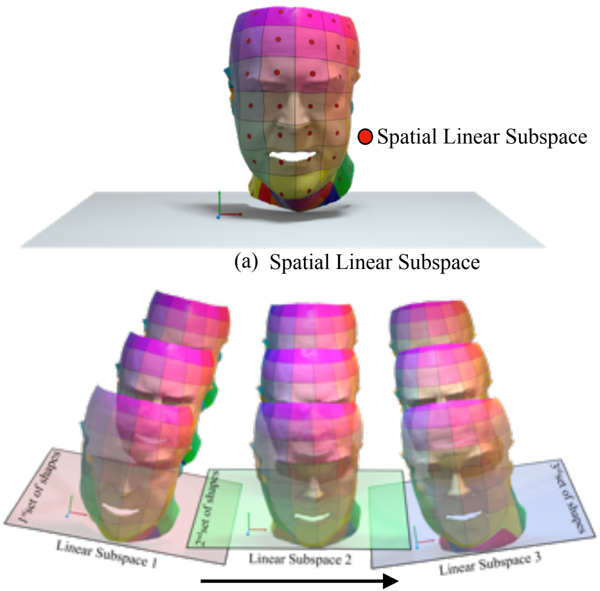
In the past, researchers have developed dense NRSfM algorithm as well, but similar to sparse NRSfM they are mostly restricted to global shape constraint [23] [3] [17]. As a result, it fails to exploit the local deformation of the surface. Moreover, the optimization framework proposed by these approaches is critically expensive to process §2. These deficiencies with past methods made us realize that a dense NRSfM algorithm needs a framework that should be able to exploit both local and global non-linearities, and at the same time must be computationally fast to process. Keeping these standards intact, we developed a new representation and modeling for dense NRSfM problem. Using our new representation, we can apply both local and global shape deformation constraints to model a dense NRSfM problem. We adhere to the assumption that the low-dimensional linear subspace spanned by a deforming shape is valid **locally** —in both space and time, along with **global** low-rank space. Such an assumption about the surfaces has been well studied in topological manifold theory [24] [25].

Global low rank representation: Matrix factorization approaches to NRSfM assumes that the deforming 3D structure intrinsically spans a low-rank space globally [5] [3]. This low-rank representation faithfully captures the global behavior of a non-rigidly deforming object over multiple frames. This representation can be obtained using Singular Value Decomposition, however, in the presence of noisy measurements such a solution can provide unsettling results. Due to [26] [27] [28] its possible to recover a low-rank solution under such circumstances. Dai et.al. research [5] —which is a classical work in NRSfM, draws its inspiration using the following optimization to solve for low-rank non-rigid 3D structure

$$\begin{aligned} &\underset{\mathtt{X},\mathtt{E}}{\operatorname{argmin}} \ \|\mathtt{X}\|_* + \|\mathtt{E}\|_l \\ &\text{subject to: } \mathtt{Y} = \mathtt{X} + \mathtt{E} \end{aligned} \tag{1}$$

Here '$\mathtt{Y}$', '$\mathtt{X}$' are the data matrix and its clean low-rank matrix representation respectively, '$\mathtt{E}$' is the error matrix and l stands for matrix norm (l = 1 or 2). Inspired by Dai et.al. [5] sparse NRSfM work, in this paper, we assume that a dense non-rigid structure is of intrinsically low rank globally. We use this assumption to capture the global deformation of the surface.

Local linear subspace representation: In contrast to the previous dense NRSfM methods [3] [17] [16], we represent deforming surfaces as a union of locally linear subspace. We argue that most non-rigidly deforming object over time and space is composed of a union of linear subspaces. Therefore, the methods that uses only global nuclear norm to constrain the surface deformation gets a solution on the convex envelop over the underlying multiple subspaces [3] [17] [16]. More precisely, these methods look for a solution on the boundaries of a feasible region which is composed of summation of subspaces. As a result, their 3D reconstruction results are empirically inferior. In this work, we aim to jointly recover an implicit local subspace representation of the surface along with its 3D reconstruction for multiple frames. Consequently, we decompose the surface into a set of locally linear subspaces and represent each subspace as a point on a Grassmann Manifold [29] (see Fig.1). We will show, such a representation is well-suited for many dense deforming surfaces. Our approximation holds well to capture the local non-linearities in addition to the global deformation.



(a) Spatial Linear Subspace

(b) Temporal Linear Subspace

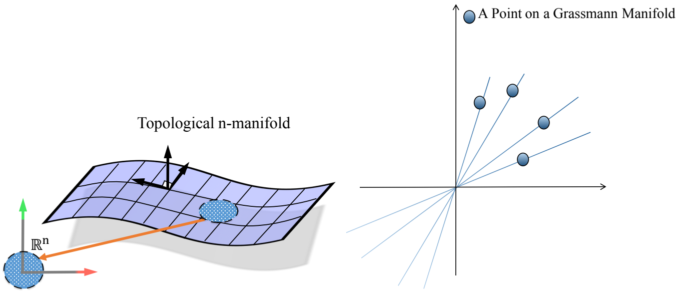
Fig. 1: Illustration show the decomposition of deforming surface into a set of local linear subspace. (a) Local Linear subspace representation of the trajectory space in the spatial domain (b) Local Linear Subspace representation of the shape space in the temporal domain. Each local plane shown in the temporal space can also be represented as a point on the Grassmann manifold. The above result show the reconstruction results using our method which is composed of 73,765 point samples from Actor dataset [30].

Our representation to assign each local linear subspace as a point on a Grassmann manifold not only provides a richer representation to exploit each local linear subspace, but also helps to improve the scalability, robustness, and processing time of our dense NRSfM algorithm. To bootstrap the initial Grassmann points, we group the local trajectories and shapes via k-means++ [31]. We compute a large but finite set of linear subspaces and, cast dense NRSfM problem as a joint clustering of subspaces and 3D reconstruction tasks. Our representation easily blends into a joint clustering and reconstruction formulation which provides superior results than performing the two tasks separately [20] [13].

Contributions: The main contributions of our work are as follows:

- A new representation for dense NRSfM problem that utilizes both local and global structure of the deforming shape to solve the problem.
- An efficient framework for modeling non-rigidly deforming surface on Grassmann manifold, which jointly supply 3D reconstruction and compact subspace representation of the shape.
- A scalable, robust and fast algorithm which does not need any template prior to solve dense NRSfM [32] [33].
- A geometry aware extension that help exploit the Grassmannian representation of different dimensions which is extremely useful in handling noise and high-dimensional Grassmannians.
- Iterative solution to the proposed optimization based on ADMM [34] that achieves leading performance on the standard benchmark dataset [3] [35].

In addition to the 3D reconstruction accuracy analysis, we performed other relevant experiments and demonstrate the advantage of our formulation using a range of qualitative and quantitative analysis. The present journal paper is based on two CVPR conference papers [36] [37]. In this work, we described the approach in greater detail including the representation, modeling of the problem and the implementation of the algorithm. Additionally, how the first proposed algorithm [36] led the foundation for the development of the next algorithm [37]. We also present a more detailed derivation of the proposed optimization with deeper statistical analysis, minor corrections and extensive experimental results. Lastly, we provide a concise discussion on the potential limitations of the algorithm, and how it can be improved further for real-world applications. We believe our journal version is much more complete, and provide the readers comprehensive details on the advantages/limitations of Grassmannian representation to solve dense NRSfM under joint 3D reconstruction and subspace clustering framework.



(a) (b)

Fig. 2: Grassmannian $\mathscr{G}(1, 2)$. (a) Intuition of n-manifold. (b) 1-dimensional Euclidean subspace of $\mathbb{R}^2$ as a point on a Grassmann Manifold. Fig.(b) is inspired from [48] work.

2 Related Work

Non-Rigid Structure from Motion (NRSfM) is more than a two-decade-old problem and is still an active area of research in the geometric computer vision. NRSfM using matrix factorization introduced in the Bregler et.al. seminal work [4] was one the first working algorithm for NRSfM, which in fact, was an extension to the rigid factorization method [38]. This problem is challenging due to the inherent unconstrained nature of the problem, as many 3D varying configurations can have similar image projections and as a result, the problem remains unsolved for any arbitrary deformations. However, many profound algorithms under some or the other prior assumptions —about the object deformation or the camera projection, have been proposed to achieve a reliable solution to this problem [5] [10] [39] [11] [20] [21] [13] [3] [40]. The literature on this topic is very extensive and therefore, for brevity, we review the works that are of close relevance to the **dense** NRSfM methods under classical NRSfM setting[2].

Earlier attempts to solve this problem used piecewise 3D reconstruction of the shape parts, which were further processed via a stitching step to get a global 3D shape [41] [42]. To our knowledge, Garg et.al. [23] variational approach was one of the first to propose and demonstrate a practical dense NRSfM algorithm that do not rely on a 3D template prior. This method introduced a discrete total variational constraint with trace norm constraint on the global shape, which leads to a biconvex optimization problem. Despite the algorithm's outstanding performance, it's computationally expensive and needs a **GPU** to provide the solution.

Recently, Dai et.al. [17] extended his simple prior free approach [5] to solve dense NRSfM problem. They proposed a spatial-temporal formulation to tackle the problem. The author revisits the temporal smoothness term from [5] 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, thus making it **extremely slow** to process, hence, not scalable for practical settings.

The consecutive frame-based formulation in recent years has shown some promising results to solve dense 3D reconstruction of a general dynamic scene, including non-rigid object [43], [44]. Nevertheless, motion segmentation, triangulation, as rigid as possible assumption, scale consistency and inter-frame consistency quite often breaks down for the deforming object. Therefore, these methods are still not mature to solve dense NRSfM for multiple frames. Not long ago, Gallardo et.al. [45] combined shading, motion and generic physical deformation to model dense NRSfM. Nevertheless, such information is generally not available on many real-world devices.

Other variants of dense non-rigid structure from motion algorithm involve solving the problem in a sequential manner *i.e.*, rather than using an entire batch of frames, solve for dense 3D reconstruction as the image arrives. However, the proposed method under such a setting uses an initial set of frames to initialize the algorithm using a rigid factorization algorithm [38], [46]. Such heuristics greatly limits the use of such a sequential method to real-world scenarios. Recently proposed method CMDR [47] proposed a hybrid approach extracts prior shape knowledge from an input sequence and, uses it as a dynamic shape prior for sequential surface recovery.

Recent state-of-the-art in sparse NRSfM uses joint subspace clustering and reconstruction formulation [13] [21]. Yet, the nature of the formulations fails to cope up with a large number of features points, and its inherent representation is unable to exploit the local surface deformation (spatially). However, the construction of simultaneous clustering and 3D reconstruction framework does provide an inspiration to extend such an idea to dense NRSfM. In this work, we want to take a step further and would like to show that it is, in fact, possible to develop the elementary idea proposed in [21] for dense NRSfM using a new representation and formulation.

3 Preliminaries

In this paper, $\|.\|_{\text{F}}$, $\|.\|_{*}$ denotes the Frobenius norm and nuclear norm respectively. $\|.\|_{\mathscr{G}}$ represents the notion of a 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 a few important definitions related to the Grassmann manifold. Firstly, a topological n-manifold ($\mathscr{M}$) is a *topological space* which is *locally homeomorphic* to a *n*-ball, where *n* is a positive integer which is well-defined, which is the dimension of the manifold. Additionally, space ($\mathscr{M}$) is assumed to be Hausdorff and second countable. Avoiding the mathematical rigor, intuitively, one can think of a continuous surface to be locally similar to the Euclidean space (see Fig. 2(a)). Out of several manifolds, the Grassmann manifold is a topologically rich non-linear manifold, each point of which represents the set of all right invariant subspace of the Euclidean space [25], [29], [36] (see Fig. 2(b)).

[2] By classical NRSfM setting, we mean the input to the algorithm is only image feature correspondences rather than depth or 3D template.

Definition 3.1. The Grassmann manifold, denoted by $\mathscr{G}(\mathtt{p},\mathtt{d})$, consists of all the linear p-dimensional subspace embedded in a 'd' dimensional Euclidean space $\mathbb{R}^{\mathtt{d}}$ such that $0 \leq \mathtt{p} \leq \mathtt{d}$ [29].

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 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 3.2. Grassmann manifold can be embedded into the space of symmetric matrices via mapping Π: $\mathscr{G}(\mathtt{p},\mathtt{d}) \mapsto$ Sym $(\mathtt{d})$, $\Pi(\Phi) = \Phi\Phi^{\mathtt{T}}$, where Φ is a Grassmann point [49], [50]. Given two Grassmann points Φ_1 and Φ_2, the distance between them can be measured using a projection metric:

$$d_g^2(\Phi_1,\Phi_2) = 0.5\|\Pi(\Phi_1) - \Pi(\Phi_2)\|_{\mathtt{F}}^2. \tag{2}$$

These two properties of Grassmann manifold has been used in many computer vision applications [49] [51] [36] [50]. Second definition is very important as it allows to measure the distance on the Grassmann manifold, hence, $(\mathscr{G}, d_g)$ forms a metric space. We used these properties in the construction of our formulation. For comprehensive details on this topic readers may refer to [49].

3.1 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 [52] [53] [20]. To handle such complex models globally can be very challenging —both numerically and computationally [3] [23]. Consequently, for an appropriate representation of such a model, we decompose the overall non-linearity of the shape by a set of locally linear models that span a low-rank subspace of a vector space. The space of all d-dimensional linear subspaces of $\mathbb{R}^{\mathtt{N}}$ $(0 < \mathtt{d} < \mathtt{N})$ forms the Grassmann manifold [24] [29]. 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 singular values and corresponding singular vectors, which makes such a representation scalable and robust to noise [54].

4 Problem Formulation

4.1 Structure and Motion Representation

Tomasi et.al. [38] matrix factorization method to represent the shape and motion under orthographic camera projection appropriately summarizes the behavior of the 3D points over frames. The relation between 3D shape, motion and its projection over frames was defined as

$$\mathtt{W} = \mathtt{RS} \tag{3}$$

where, $\mathtt{W} \in \mathbb{R}^{2\mathtt{F}\times\mathtt{P}}$ is the measurement matrix with 'P' as the total number of feature points tracked across 'F' frames. $\mathtt{R} = \mathtt{blockdiagonal}(\mathtt{R}_1,\mathtt{R}_2,..,\mathtt{R}_{\mathtt{F}}) \in \mathbb{R}^{2\mathtt{F}\times3\mathtt{F}}$ denotes the orthographic camera rotation matrix with each $\mathtt{R}_{\mathtt{i}} \in \mathbb{R}^{2\times3}$ as per frame rotation. $\mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}}$ represent the shape matrix with each row triplet as a 3D shape. This representation was originally formulated to solve **rigid** structure from motion under orthographic projection which was later extended by [4] to recover the 3D shape of a **non-rigidly** deforming object for multiple frames.

This classical representation entails that given the input measurement matrix, solve for rotation $(\mathtt{R})$ and 3D shape $(\mathtt{S})$. For our method, we solve for rotation matrix using the Intersection method [5] by assuming that per frame relative camera motion $(\mathtt{R})$ can faithfully represent the global deformation of the subject in the scene. Accordingly, our goal reduces to develop a systematic approach that can reliably explain the non-rigid shape deformations and provides better 3D reconstruction. We used this relation to enforce our first constraint to solve for shape. This constraint is also known as a **re-projection error constraint** *i.e,*

$$\underset{\mathtt{S}}{\text{minimize}}\ \frac{1}{2}\|\mathtt{W} - \mathtt{RS}\|_{\mathtt{F}}^2 \tag{4}$$

4.2 Non-Rigid Object Representation

As alluded to above, given the matrix $\mathtt{R}$, our goal is to solve for the 3D structure $\mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}}$. Eq:(3) show that we can get infinite family of solution to $\mathtt{S}$ using such representation. Nonetheless, Bregler et.al. [4] $3\mathtt{K}$ matrix factorization to obtain low-order linear model suggest that $\mathtt{rank}(\mathtt{S}) \leq 3\mathtt{K}$. Consequently, the non-rigid 3D shape must lie in a low-rank space. Later, Akther et.al. [2] and Dai et.al. [5] suggested the idea to provide stronger rank bound to the shape matrix by shuffling the arrangements of rows and columns of shape matrix *i.e*, $\mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}} \mapsto \mathtt{S}^{\sharp} \in \mathbb{R}^{3\mathtt{P}\times\mathtt{F}}$. Accordingly, we enforce the low-rank constraint on $\mathtt{S}^{\sharp}$ which equivalently represent the low order constraint [4] [2] [5] with a tighter rank bound $\mathtt{rank}(\mathtt{S}^{\sharp}) \leq \mathtt{K}$. Combining re-projection error constraint with the low-rank constraint, we have

$$\begin{aligned} &\underset{\mathtt{S},\mathtt{S}^{\sharp}}{\text{minimize}}\ \frac{1}{2}\|\mathtt{W} - \mathtt{RS}\|_{\mathtt{F}}^2 + \gamma\|\mathtt{S}^{\sharp}\|_* \\ &\text{subject to: } \mathtt{S}^{\sharp} = f(\mathtt{S}) \end{aligned} \tag{5}$$

where $\|\mathtt{S}^{\sharp}\|_*$ represent the nuclear norm of the shape matrix. Here, we define $f: \mathtt{S} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}} \mapsto \mathtt{S}^{\sharp} \in \mathbb{R}^{3\mathtt{P}\times\mathtt{F}}$. In general, the exact rank minimization problem is NP-hard, hence, we relax this with a nuclear norm minimization problem. Dai et.al. [5] proposed this formulation to solve non-rigid structure from motion problem. Although this formulation provides a decent result for sparse feature points, it fails to estimate dense non-rigid structure from motion with reasonable accuracy. One of the main reasons for its failure is that a non-rigid deforming surface is mostly composed of a union of several local linear subspaces. Consequently, a global low-rank shape constraint fails to cater the local shape deformation, therefore, it provides questionable 3D reconstruction results for a dense deforming object.

To overcome this limitation with [5] formulation, joint subspace clustering and reconstruction methods are proposed [13] [21]. Although the method proposed by [13] [21] provides state-of-the-art results for sparse features points [22], the algorithm cannot process a large set of feature points, hence, not scalable. To come up with an algorithm that is scalable and also utilize the idea of spatial-temporal clustering approach for dense non-rigid surfaces, we use grassmannian representation in our formulation [24] [55].

4.3 Grassmannian Representation in Trajectory Space

Let '$\Phi_{\mathtt{si}}$' $\in \mathscr{G}(\mathtt{p},\mathtt{d})$ be a Grassmann point representing the $\mathtt{i}^{\text{th}}$ local linear subspace spanned by $\mathtt{i}^{\text{th}}$ set of columns of 'S'. Using this notion, we decompose the entire trajectories of the structure into a set of '$\mathtt{K}_{\mathtt{s}}$' Grassmannians $\xi_{\mathtt{s}} = \{\Phi_{\mathtt{s}1},\Phi_{\mathtt{s}2},\Phi_{\mathtt{s}3},....,\Phi_{\mathtt{sK}_{\mathtt{s}}}\}$. To explain the complex non-rigid deformations, we reduce the overall non-linear space as a union of several local low-dimensional linear spaces which are 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 [56], we embed the Grassmann manifold into the symmetric matrix manifold, where the self-expressiveness can be defined in the embedding space. Let '$\mathcal{X}_s$' be a tensor which is constructed by mapping trajectory space Grassmann points. Concretely, $\mathcal{X}_s = \{(\Phi_{s1}\Phi_{s1}^T), (\Phi_{s2}\Phi_{s2}^T), ..., (\Phi_{Ks}\Phi_{Ks}^T)\}$ is its embedding onto symmetric matrix manifold which is constructed by mapping trajectory space Grassmann points. Since the high-dimensional complex deformation is composed of several low-dimensional subspace, its low rank representation shall reveal the subspace information. This motivation leads to the following optimization in the trajectory space

$$\begin{aligned} &\underset{E_s, C_s}{\text{minimize}} \ \|E_s\|_F^2 + \lambda_1 \|C_s\|_* \\ &\text{subject to: } \mathcal{X}_s = \mathcal{X}_s C_s + E_s \end{aligned} \tag{6}$$

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 subspace 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. $\| \ \|_*$ is enforced on C_s for a low-rank solution.

4.4 Grassmannian Representation in Shape Space

Similarly, let 'Φ_{ti}' $\in \mathscr{G}(p, d)$ be a Grassmann point representing the i^{th} local linear subspace spanned by i^{th} set of columns of '$S^\sharp$'. We decompose the set of shapes into 'K_t' Grassmannians $\xi_t = \{\Phi_{t1}, \Phi_{t2}, \Phi_{t3}, ..., \Phi_{tK_t}\}$. To accomplish the notion of self-expressiveness in the temporal space as well, we define $\mathcal{X}_t = \{(\Phi_{t1}\Phi_{t1}^T), (\Phi_{t2}\Phi_{t2}^T), ..., (\Phi_{Kt}\Phi_{Kt}^T)\}$. Previous literature and experiments revealed that deforming object attains different state over time which adheres to distinct temporal local linear subspaces [21]. Assuming that the temporal deformation is smooth over-time, we express deforming shapes in terms of local self-expressiveness on grassmann manifold across frames as:

$$\begin{aligned} &\underset{E_t, C_t}{\text{minimize}} \ \|E_t\|_F^2 + \lambda_2 \|C_t\|_* \\ &\text{subject to: } \mathcal{X}_t = \mathcal{X}_t C_t + E_t \end{aligned} \tag{7}$$

where, $\mathcal{X}_t$ is the set of all symmetric matrices constructed using a set of Grassmannian samples ξ_t, where ξ_t contains the samples which are obtained from $S_t^\sharp \in \mathbb{R}^{3P \times F}$. Intuitively, $S_t^\sharp$ 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. $\| \ \|_*$ is enforced on C_t for a low-rank solution.

4.5 Simplified Low Rank Subspace Representation on the Grassmann Manifold

The grassmannian representation in Eq.(6) and Eq.(7) are not straight-forward to solve, we simplified it further to an equivalent optimization problem which is easy to optimize. Consider the following minimization problem

$$\underset{C}{\text{minimize}} \ \|\mathcal{X} - \mathcal{X}C\|_F^2 + \lambda \|C\|_* \tag{8}$$

Here, we choose the notations that stands for both Eq.(6) and Eq.(7). To simplify the form of previous optimization problem, let's consider the error term that involves the tensor structure.

$$\|E\|_F^2 = \|\mathcal{X} - \mathcal{X}C\|_F^2 \tag{9}$$

Using our notation $\mathcal{X} = \{(\Phi_1\Phi_1^T), (\Phi_2\Phi_2^T), ..., (\Phi_N\Phi_N^T)\}$ and $C \in \mathbb{R}^{N \times N}$ are the set of grassmann samples in the embedding space and its coefficient matrix respectively. Let's re-write the previous equation

$$\|E\|_F^2 = \sum_{i=1}^{N} \|E_i\|_F^2 = \sum_{i=1}^{N} \mathbf{Tr}(E_i^T E_i) \tag{10}$$

Using the per sample notion *i.e,* any sample can be represented as a combination of other samples in the same space.

$$E_i = \Phi_i\Phi_i^T - \sum_{j=1}^{N} c_{ij}(\Phi_j\Phi_j^T) \tag{11}$$

Substituting the above expression in Eq.(10) for the i^{th} sample, we write

$$\begin{aligned} \|E_i\|_F^2 &= \mathbf{Tr}(E_i^T E_i) \\ &= \mathbf{Tr}\Big[\Big(\Phi_i\Phi_i^T - \sum_{j=1}^{N} c_{ij}(\Phi_j\Phi_j^T)\Big)^T\Big(\Phi_i\Phi_i^T - \sum_{j=1}^{N} c_{ij}(\Phi_j\Phi_j^T)\Big)\Big] \end{aligned} \tag{12}$$

Expanding the above form

$$\begin{aligned} \|E_i\|_F^2 &= \mathbf{Tr}\big((\Phi_i\Phi_i^T)^T(\Phi_i\Phi_i^T)\big) - 2\sum_{j=1}^{N} c_{ij}\mathbf{Tr}\big((\Phi_i\Phi_i^T)^T(\Phi_j\Phi_j^T)\big) \\ &+ \sum_{l=1}^{N}\sum_{m=1}^{N} c_{il}c_{im}\mathbf{Tr}\big((\Phi_l\Phi_l^T)^T(\Phi_m\Phi_m^T)\big) \end{aligned} \tag{13}$$

Using the cyclic trace property and the orthonormality property of matrices (**Definition** 3.1).

$$\begin{aligned} \|E_i\|_F^2 &= \mathbf{Tr}(\mathbf{I}_p) - 2\sum_{j=1}^{N} c_{ij}\mathbf{Tr}\big((\Phi_j^T\Phi_i)(\Phi_i^T\Phi_j)\big) \\ &+ \sum_{l=1}^{N}\sum_{m=1}^{N} c_{il}c_{im}\mathbf{Tr}\big((\Phi_l^T\Phi_m)(\Phi_m^T\Phi_m)\big) \end{aligned} \tag{14}$$

Here p is the magnitude of the lower dimensional space representation (**Definition** 3.1). By letting $\Gamma_{ij} = \mathbf{Tr}\big((\Phi_j^T\Phi_i)(\Phi_i^T\Phi_j)\big)$, we can rewrite the above form as

$$\|E_i\|_F^2 = \mathbf{Tr}(\mathbf{I}_p) - 2\sum_{j=1}^{N} c_{ij}\Gamma_{ij} + \sum_{l=1}^{N}\sum_{m=1}^{N} c_{il}c_{im}\Gamma_{lm} \tag{15}$$

Notice that Γ_{ij} is $\mathbb{R}^{p \times p}$ matrix which is much easier to compute and process. Also, it's simple to verify that Γ_{ij} is symmetric. Let $\Gamma = (\Gamma_{ij})_{ij=1}^{N} \in \mathbb{R}^{N \times N}$. By summing over all the samples , we can rewrite Eq:(10) as:

$$\begin{aligned} \|E\|_F^2 &= Np - 2\mathbf{Tr}(C\Gamma) + \mathbf{Tr}(C\Gamma C^T) \\ &\equiv Np - 2\mathbf{Tr}(CLL^T) + \mathbf{Tr}(CLL^TC^T) \\ &\equiv \text{const} + \|L - CL\|_F^2 \end{aligned} \tag{16}$$

where, $LL^T = \mathbf{Chol}(\Gamma)$, the Cholesky decomposition of the matrix. Note that adding and subtracting constant symbol w.r.t variable C will not affect the solution to the targeted optimization problem. Using this form, we simplify the Eq:(8) optimization problems as

$$\underset{C}{\text{minimize}} \ \|L - CL\|_F^2 + \lambda \|C\|_* \tag{17}$$

This simplified equivalent problem is much easier to solve and process. We will use this form in our overall cost function to solve non-rigid 3D shape reconstruction.

5 Spatial Temporal Formulation

Combining the above developed objectives and their constraints give us our spatial temporal formulation for dense NRSfM. Our representation blends the local subspaces structure along with the global composition of a non-rigid shape. Thus, the overall objective is:

$$\begin{aligned}
&\underset{\mathtt{S},\mathtt{S}^{\sharp},\mathtt{E}_{\mathtt{s}},\mathtt{E}_{\mathtt{t}},\mathtt{C}_{\mathtt{s}},\mathtt{C}_{\mathtt{t}}}{\text{minimize}} \ \mathbf{E} = \frac{1}{2}\ \|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2 + \mu\|\mathtt{S}^{\sharp}\|_* + \lambda_1\|\mathtt{E}_{\mathtt{s}}\|_{\mathtt{F}}^2 + \lambda_2\|\mathtt{E}_{\mathtt{t}}\|_{\mathtt{F}}^2 \\
&+\lambda_3\|\mathtt{C}_{\mathtt{s}}\|_* + \lambda_4\|\mathtt{C}_{\mathtt{t}}\|_* \\
&\text{subject to :} \\
&\mathcal{X}_{\mathtt{s}} = \mathcal{X}_{\mathtt{s}}\mathtt{C}_{\mathtt{s}} + \mathtt{E}_{\mathtt{s}};\ \mathcal{X}_{\mathtt{t}} = \mathcal{X}_{\mathtt{t}}\mathtt{C}_{\mathtt{t}} + \mathtt{E}_{\mathtt{t}}; \\
&\xi_{\mathtt{s}} = f_g(\mathbf{P}_{\mathtt{s}},\mathtt{S},\mathtt{K}_{\mathtt{s}},\mathtt{p}_{\mathtt{s}});\ \xi_{\mathtt{t}} = f_g(\mathbf{P}_{\mathtt{t}},\mathtt{S}^{\sharp},\mathtt{K}_{\mathtt{t}},\mathtt{p}_{\mathtt{t}}); \\
&\mathtt{S} = f_s(\xi_{\mathtt{s}},\Sigma_{\mathtt{s}},\xi_{\mathtt{vs}},\mathtt{K}_{\mathtt{s}},\mathtt{p}_{\mathtt{s}});\ \mathtt{S}^{\sharp} = f_s(\xi_{\mathtt{t}},\Sigma_{\mathtt{t}},\xi_{\mathtt{vt}},\mathtt{K}_{\mathtt{t}},\mathtt{p}_{\mathtt{t}}); \\
&\mathbf{P}_{\mathtt{s}} = f_p(\xi_{\mathtt{s}},\mathtt{C}_{\mathtt{s}},\mathbf{P}_{\mathtt{so}});\ \mathbf{P}_{\mathtt{t}} = f_p(\xi_{\mathtt{t}},\mathtt{C}_{\mathtt{t}},\mathbf{P}_{\mathtt{to}}); \\
&\mathtt{S}^{\sharp} = f(\mathtt{S});\ \mathtt{W} = f_o(\mathtt{W},\mathbf{P}_{\mathtt{s}});
\end{aligned} \tag{18}$$

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

$$\begin{aligned}
&\Big\{\big(\xi, f_g(\mathbf{P},\mathtt{X},\mathtt{K},\mathtt{p})\big) : \text{order } \{\mathtt{X}_{\mathtt{i}}\}_{\mathtt{i}=1}^{\mathtt{K}} \text{ columns of } \mathtt{X} \text{ using } \mathbf{P}, \\
&\xi := \big(\Phi_{\mathtt{i}}\big|_{\mathtt{i}=1}^{\mathtt{K}}\big),\ \text{where}, [\Phi_{\mathtt{i}},\Sigma_{\mathtt{i}},\xi_{\mathrm{vi}}^{\mathtt{T}}] = \text{svds}(\mathtt{X}_{\mathtt{i}},\mathtt{p})\Big\}
\end{aligned} \tag{19}$$

$$\begin{aligned}
&\Big\{\big(\mathtt{X}, f_s(\xi,\Sigma,\xi_{\mathtt{v}},\mathtt{K},\mathtt{p})\big) : \mathtt{X}_{\mathtt{i}} = [\xi_{\mathtt{i}}\ \Sigma_{\mathtt{i}}\ \xi_{\mathtt{vi}}^{\mathtt{T}}], \text{where } \Sigma_{\mathtt{i}} \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}, \\
&\mathtt{X} = \big(\mathtt{X}_{\mathtt{i}}\big|_{\mathtt{i}=1}^{\mathtt{K}}\big)\Big\}
\end{aligned} \tag{20}$$

$$\Big\{(\mathbf{P}, f_p(\xi,\mathtt{C},\mathbf{P}_{\mathtt{o}}) : \mathbf{P} = \text{spectral\_clustering}(\xi,\mathtt{C},\mathbf{P}_{\mathtt{o}})\Big\} \tag{21}$$

$$\Big\{(\mathtt{X}^{\sharp}, f(\mathtt{X}) : \mathtt{X} \in \mathbb{R}^{3\mathtt{F}\times\mathtt{P}} \mapsto \mathtt{X}^{\sharp} \in \mathbb{R}^{3\mathtt{P}\times\mathtt{F}}\Big\} \tag{22}$$

$$\Big\{(\mathtt{X}, f_o(\mathtt{X},\mathbf{P})) : \mathtt{X} = \text{arrange columns of } \mathtt{X} \text{ using ordering vector } \mathbf{P}\Big\} \tag{23}$$

Note: $\big(\mathtt{X}_{\mathtt{i}}\big|_{\mathtt{i}}^{\mathtt{K}}\big)$ denoted the horizontal concatenation of the matrices. Intuitively, $f_g(.)$ provides the Grassmannian representation and $f_s(.)$ reconstructs back each local low-rank subspace. $f_p(.)$ provides the ordering vector based on the inference drawn from coefficient matrix and f_o rearranges the columns of $\mathtt{W}$ in accordance with the columns of the shape matrix. The proposed cost function is minimized by solving for one variable at a time while treating others as constant, keeping the constraints intact over iteration. Next, we provide a detailed derivation to each sub-problem.

5.1 Solution

The formulation in Eq.(18) is a non-convex problem due to the bilinear optimization variables ($\mathcal{X}_{\mathtt{s}}\mathtt{C}_{\mathtt{s}}$, $\mathcal{X}_{\mathtt{t}}\mathtt{C}_{\mathtt{t}}$), hence a global optimal solution is hard to achieve. However, it can be efficiently solved using Augmented Lagrangian Methods (ALMs) [34], which has proven its effectiveness for many non-convex problems. Using the result of Eq.(17) with introduction of Lagrange multipliers ($\{\mathbf{L}_{\mathtt{i}}\}_{\mathtt{i}=1}^{3}$) and auxiliary variables ($\mathtt{J}_{\mathtt{s}}, \mathtt{J}_{\mathtt{t}}$) to Eq. (18) gives us the overall cost function as follows:

$$\begin{aligned}
&\underset{\mathtt{S},\mathtt{S}^{\sharp},\mathtt{J}_{\mathtt{s}},\mathtt{J}_{\mathtt{t}},\mathtt{C}_{\mathtt{s}},\mathtt{C}_{\mathtt{t}}}{\text{minimize}} \ \mathbf{E} = \frac{1}{2}\ \|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{S}^{\sharp}-f(\mathtt{S})\|_F^2 + <\mathbf{L}_1, \mathtt{S}^{\sharp}-f(\mathtt{S})> \\
&\gamma\|\mathtt{S}^{\sharp}\|_* + \lambda_1\|\mathtt{L}_{\mathtt{s}}-\mathtt{C}_{\mathtt{s}}\mathtt{L}_{\mathtt{s}}\|_{\mathtt{F}}^2 + \lambda_3\|\mathtt{J}_{\mathtt{s}}\|_* + \lambda_2\|\mathtt{L}_{\mathtt{t}}-\mathtt{C}_{\mathtt{t}}\mathtt{L}_{\mathtt{t}}\|_{\mathtt{F}}^2 + \lambda_4\|\mathtt{J}_{\mathtt{t}}\|_* + \\
&\frac{\beta}{2}\|\mathtt{C}_{\mathtt{s}}-\mathtt{J}_{\mathtt{s}}\|_{\mathtt{F}}^2 + <\mathbf{L}_2, \mathtt{C}_{\mathtt{s}}-\mathtt{J}_{\mathtt{s}}> + \frac{\beta}{2}\|\mathtt{C}_{\mathtt{t}}-\mathtt{J}_{\mathtt{t}}\|_{\mathtt{F}}^2 + <\mathbf{L}_3, \mathtt{C}_{\mathtt{t}}-\mathtt{J}_{\mathtt{t}}> \\
&\text{subject to :} \\
&\xi_{\mathtt{s}} = f_g(\mathbf{P}_{\mathtt{s}},\mathtt{S},\mathtt{K}_{\mathtt{s}},\mathtt{p}_{\mathtt{s}});\ \xi_{\mathtt{t}} = f_g(\mathbf{P}_{\mathtt{t}},\mathtt{S}^{\sharp},\mathtt{K}_{\mathtt{t}},\mathtt{p}_{\mathtt{t}}); \\
&\mathtt{S} = f_s(\xi_{\mathtt{s}},\Sigma_{\mathtt{s}},\xi_{\mathtt{vs}},\mathtt{p}_{\mathtt{s}},\mathtt{K}_{\mathtt{s}});\ \mathtt{S}^{\sharp} = f_s(\xi_{\mathtt{t}},\Sigma_{\mathtt{t}},\xi_{\mathtt{vt}},\mathtt{p}_{\mathtt{t}},\mathtt{K}_{\mathtt{t}}); \\
&\mathbf{P}_{\mathtt{s}} = f_p(\xi_{\mathtt{s}},\mathtt{C}_{\mathtt{s}},\mathbf{P}_{\mathtt{so}});\ \mathbf{P}_{\mathtt{t}} = f_p(\xi_{\mathtt{t}},\mathtt{C}_{\mathtt{t}},\mathbf{P}_{\mathtt{to}}); \\
&\mathtt{S}^{\sharp} = f(\mathtt{S});\ \mathtt{W} = f_o(\mathtt{W},\mathbf{P}_{\mathtt{s}});
\end{aligned} \tag{24}$$

Solution to $\mathtt{S}$

$$\begin{aligned}
&\underset{\mathtt{S}}{\text{argmin}}\frac{1}{2}\ \|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{S}^{\sharp}-f(\mathtt{S})\|_{\mathtt{F}}^2 + <\mathbf{L}_1, \mathtt{S}^{\sharp}-f(\mathtt{S})> \\
&\underset{\mathtt{S}}{\text{argmin}}\frac{1}{2}\ \|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|f^{-1}(\mathtt{S}^{\sharp})-\mathtt{S}\|_{\mathtt{F}}^2 + <f^{-1}(\mathbf{L}_1), f^{-1}(\mathtt{S}^{\sharp})-\mathtt{S}> \\
&\equiv \underset{\mathtt{S}}{\text{argmin}}\frac{1}{2}\ \|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{S}-\big(f^{-1}(\mathtt{S}^{\sharp}) + \frac{f^{-1}(\mathbf{L}_1)}{\beta}\big)\|_{\mathtt{F}}^2.
\end{aligned} \tag{25}$$

The solution to the variable '$\mathtt{S}$' can be derived by differentiating the above term w.r.t $\mathtt{S}$ and equating it to zero.

$$\mathtt{S} \equiv \Big(\mathtt{R}^{\mathtt{T}}\mathtt{R} + \beta\mathtt{I}\Big)^{-1}\Big(\beta\big(f^{-1}(\mathtt{S}^{\sharp}) + \frac{f^{-1}(\mathbf{L}_1)}{\beta}\big) + \mathtt{R}^{\mathtt{T}}\mathtt{W}\Big) \tag{26}$$

Solution to $\mathtt{S}^{\sharp}$

$$\begin{aligned}
&\equiv \underset{\mathtt{S}^{\sharp}}{\text{argmin}}\ \gamma\|\mathtt{S}^{\sharp}\|_* + \frac{\beta}{2}\|S^{\sharp}-f(\mathtt{S})\|_{\mathtt{F}}^2 + <\mathbf{L}_1, \mathtt{S}^{\sharp}-f(\mathtt{S})> \\
&\equiv \underset{\mathtt{S}^{\sharp}}{\text{argmin}}\ \gamma\|\mathtt{S}^{\sharp}\|_* + \frac{\beta}{2}\|\mathtt{S}^{\sharp}-\big(f(\mathtt{S}) - \frac{\mathbf{L}_1}{\beta}\big)\|_F^2
\end{aligned} \tag{27}$$

Let's define the soft-thresholding operation as $\mathscr{S}_{\tau}[x] = \text{sign}(x)$ $\max(|x|-\tau, 0)$. The optimal solution to $\mathtt{S}^{\sharp}$ is given by

$$\mathtt{S}^{\sharp} \equiv \mathtt{U}\mathscr{S}_{\frac{\gamma}{\beta}}(\Sigma)\mathtt{V}^{\mathtt{T}} \text{ where, } [\mathtt{U},\Sigma,\mathtt{V}^{\mathtt{T}}] = \text{svd}(f(\mathtt{S}) - \frac{\mathbf{L}_1}{\beta}) \tag{28}$$

Solution to $\mathtt{C}_{\mathtt{s}}$

$$\begin{aligned}
&\equiv \underset{\mathtt{C}_{\mathtt{s}}}{\text{argmin}}\ \lambda_1\|\mathtt{L}_{\mathtt{s}}-\mathtt{C}_{\mathtt{s}}\mathtt{L}_{\mathtt{s}}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{C}_{\mathtt{s}}-\mathtt{J}_{\mathtt{s}}\|_{\mathtt{F}}^2 + <\mathbf{L}_2, \mathtt{C}_{\mathtt{s}}-\mathtt{J}_{\mathtt{s}}> \\
&\equiv \underset{\mathtt{C}_{\mathtt{s}}}{\text{argmin}}\ \lambda_1\|\mathtt{L}_{\mathtt{s}}-\mathtt{C}_{\mathtt{s}}\mathtt{L}_{\mathtt{s}}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{C}_{\mathtt{s}}-\big(\mathtt{J}_{\mathtt{s}} - \frac{\mathbf{L}_2}{\beta}\big)\|_{\mathtt{F}}^2
\end{aligned} \tag{29}$$

The solution to $\mathtt{C}_{\mathtt{s}}$ can be derived by differentiating the above term w.r.t $\mathtt{C}_{\mathtt{s}}$ and equating it to zero.

$$\mathtt{C}_{\mathtt{s}} \equiv \Big(2\lambda_1\mathtt{L}_{\mathtt{s}}\mathtt{L}_{\mathtt{s}}^{\mathtt{T}} + \beta(\mathtt{J}_{\mathtt{s}} - \frac{\mathbf{L}_2}{\beta})\Big)\Big(2\lambda_1\mathtt{L}_{\mathtt{s}}\mathtt{L}_{\mathtt{s}}^{\mathtt{T}} + \beta\mathtt{I}_{\mathtt{s}}\Big)^{-1} \tag{30}$$

Solution to $\mathtt{C}_{\mathtt{t}}$

Similar to the $\mathtt{C}_{\mathtt{s}}$ derivation, it's solution can be derived as follows:

$$\begin{aligned}
&\equiv \underset{\mathtt{C}_{\mathtt{t}}}{\text{argmin}}\ \lambda_2\|\mathtt{L}_{\mathtt{t}}-\mathtt{C}_{\mathtt{t}}\mathtt{L}_{\mathtt{t}}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{C}_{\mathtt{t}}-\mathtt{J}_{\mathtt{t}}\|_{\mathtt{F}}^2 + <\mathbf{L}_3, \mathtt{C}_{\mathtt{t}}-\mathtt{J}_{\mathtt{t}}> \\
&\equiv \underset{\mathtt{C}_{\mathtt{t}}}{\text{argmin}}\ \lambda_2\|\mathtt{L}_{\mathtt{t}}-\mathtt{C}_{\mathtt{t}}\mathtt{L}_{\mathtt{t}}\|_{\mathtt{F}}^2 + \frac{\beta}{2}\|\mathtt{C}_{\mathtt{t}}-\big(\mathtt{J}_{\mathtt{t}} - \frac{\mathbf{L}_3}{\beta}\big)\|_F^2
\end{aligned} \tag{31}$$

$$\mathtt{C}_{\mathtt{t}} \equiv \Big(2\lambda_2\mathtt{L}_{\mathtt{t}}\mathtt{L}_{\mathtt{t}}^{\mathtt{T}} + \beta(\mathtt{J}_{\mathtt{t}} - \frac{\mathbf{L}_3}{\beta})\Big)\Big(2\lambda_2\mathtt{L}_{\mathtt{t}}\mathtt{L}_{\mathtt{t}}^{\mathtt{T}} + \beta\mathtt{I}_{\mathtt{t}}\Big)^{-1} \tag{32}$$

Algorithm 1 Dense Non-Rigid Structure from Motion using Grassmannians

```
Require: W, R using [5], tuning parameters: $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$, $\gamma$, $\rho = 1.1$, $\beta = 1e^{-3}$, $\beta_m = 1e^{6}$, $\varepsilon = 1e^{-12}$, $K_s$, $K_t$.
  Initialize: $S = \mathbf{pinv}(R)W$ and $S^\sharp = f(S)$.
  Initialize: '$K_t$' temporal data points on the Grassmann manifold using $\mathbf{P}_{to}$ = kmeans++($S^\sharp, K_t$) index to '$S^\sharp$' matrix, $\xi_t = \{\Phi_{ti}\}_{i=1}^{K_t}$
  Initialize: '$K_s$' spatial data points on the Grassmann manifold using $\mathbf{P}_{so}$ = kmeans++($S, K_s$) index to '$S$' matrix, $\xi_s = \{\Phi_{si}\}_{i=1}^{K_s}$
  Initialize: The auxiliary variables $J_s$, $J_t$ and Lagrange multiplier $\{\mathbf{L}_i\}_{i=1}^{3}$ as zero matrices.
  Initialize: $\Gamma_{ij}^{s} = \mathbf{Tr}[(\Phi_{sj}^T\Phi_{si})(\Phi_{si}^T\Phi_{sj})]$, $\Gamma_{ij}^{t} = \mathbf{Tr}[(\Phi_{tj}^T\Phi_{ti})(\Phi_{ti}^T\Phi_{tj})]$, $\Gamma_s = (\Gamma_{ij}^{s})_{i,j=1}^{K_s}, \Gamma_t = (\Gamma_{ij}^{t})_{i,j=1}^{K_t}$
              $L_sL_s^T = \mathbf{Chol}(\Gamma_s)$, $L_tL_t^T = \mathbf{Chol}(\Gamma_t)$
  Initialize: $\mathbf{P}_s := \mathbf{P}_{so}$, $\mathbf{P}_t := \mathbf{P}_{to}$, iter $= 1$
  Define:   $\mathscr{S}_\tau(x) := @(x,\tau)\mathrm{sign}(x).*\max(\mathrm{abs}(x)-\tau, 0)$;        {MATLAB function script}

while not converged do
    $S \leftarrow \left(R^TR + \beta I\right)^{-1}\left(\beta\left(f^{-1}(S^\sharp) + \frac{f^{-1}(\mathbf{L}_1)}{\beta}\right) + R^TW\right)$
    $C_s \leftarrow \left(2\lambda_1 L_sL_s^T + \beta(J_s - \frac{\mathbf{L}_2}{\beta})\right)\left(2\lambda_1 L_sL_s^T + \beta I_s\right)^{-1}$
    $\xi_s \leftarrow f_g(\mathbf{P}_s, S, K_s, p_s)$;        {Update spatial Grassmann points}
    $S \leftarrow f_s(\xi_s, \Sigma_s, \xi_{vs}, K_s, p_s)$;        {Refine based on top $p_s$ singular values}
    $J_s \leftarrow U_{J_s}\mathscr{S}_{\frac{\lambda_3}{\beta}}(\Sigma_{J_s})V_{J_s}^T$, where $[U_{J_s}, \Sigma_{J_s}, V_{J_s}^T] = \mathrm{svd}(C_s + \frac{\mathbf{L}_2}{\beta})$
    $S^\sharp \leftarrow U\mathscr{S}_{\frac{\gamma}{\beta}}(\Sigma)V^T$where, $[U, \Sigma, V^T] = \mathrm{svd}(f(S) - \frac{\mathbf{L}_1}{\beta})$
    $C_t \leftarrow \left(2\lambda_2 L_tL_t^T + \beta(J_t - \frac{\mathbf{L}_3}{\beta})\right)\left(2\lambda_2 L_tL_t^T + \beta I_t\right)^{-1}$
    $\xi_t \leftarrow f_g(\mathbf{P}_t, S^\sharp, K_t, p_t)$        {Update temporal Grassmann points}
    $S^\sharp \leftarrow f_s(\xi_t, \Sigma_t, \xi_{vt}, K_t, p_t)$;        {Refine based on top $p_t$ singular value}
    $J_t \leftarrow U_{J_t}\mathscr{S}_{\frac{\lambda_4}{\beta}}(\Sigma_{J_t})V_{J_t}^T$,where $[U_{J_t}, \Sigma_{J_t}, V_{J_t}^T] = \mathrm{svd}(C_t + \frac{\mathbf{L}_3}{\beta})$
    $\Gamma_{ij}^{s} \leftarrow \mathbf{Tr}[(\Phi_{sj}^T\Phi_{si})(\Phi_{si}^T\Phi_{sj})]$, $\Gamma_{ij}^{t} \leftarrow \mathbf{Tr}[(\Phi_{tj}^T\Phi_{ti})(\Phi_{ti}^T\Phi_{tj})]$;
    $\Gamma_s \leftarrow (\Gamma_{ij}^{s})_{i,j=1}^{K_s}, \Gamma_t \leftarrow (\Gamma_{ij}^{t})_{i,j=1}^{K_t}$;        {$\Gamma_s \succeq 0, \Gamma_t \succeq 0$, if $\Gamma_s || \Gamma_t = 0$ add $\delta\mathbf{I}$ to make it $\succ 0$ }
    $L_sL_s^T = \mathbf{Chol}(\Gamma_s)$, $L_tL_t^T = \mathbf{Chol}(\Gamma_t)$;
    $\mathbf{P}_s = f_p(\xi_s, C_s, \mathbf{P}_s)$; $\mathbf{P}_t = f_p(\xi_t, C_t, \mathbf{P}_t)$;
    $W \leftarrow f_o(W, \mathbf{P}_s)$        {Note: Column ordering of W and S must be same.}
    $\mathbf{L}_1 := \mathbf{L}_1 + \beta(S^\sharp - f(S))$, $\mathbf{L}_2 := \mathbf{L}_2 + \beta(C_s - J_s)$, $\mathbf{L}_3 := \mathbf{L}_3 + \beta(C_t - J_t)$; {Update Lagrange multipliers}
    $\beta \leftarrow \min(\rho\beta, \beta_m)$
    maxgap := $\max([\|S^\sharp - f(S)\|_\infty, \|C_s - J_s\|_\infty, \|C_t - J_t\|_\infty])$
    if (maxgap $< \varepsilon \vee \beta > \beta_m$) then
        break;
    end if        {Check for the convergence}
    iter := iter + 1
end while        {Note: $\delta$ is a very small positive number and $\mathbf{I}$ symbolizes identity matrix.}
Ensure: $S$, $S^\sharp$, $C_s$, $C_t$.        {Note: Kindly use economical version of 'svd()' on a regular desktop.}
```

Solution to J_s

$$\begin{aligned} &\equiv \underset{J_s}{\mathrm{argmin}}\ \lambda_3\|J_s\|_* + \frac{\beta}{2}\|C_s - J_s\|_F^2 + <\mathbf{L}_2, C_s - J_s> \\ &\equiv \underset{J_s}{\mathrm{argmin}}\ \lambda_3\|J_s\|_* + \frac{\beta}{2}\|J_s - \big(C_s + \frac{\mathbf{L}_2}{\beta}\big)\|_F^2 \end{aligned} \tag{33}$$

Similar to Eq.(28) derivation, we use the soft-thresholding operation. It's optimal solution can be obtained as

$$J_s \equiv U_{J_s}\mathscr{S}_{\frac{\lambda_3}{\beta}}(\Sigma_{J_s})V_{J_s}^T,\ \text{where}\ [U_{J_s}, \Sigma_{J_s}, V_{J_s}^T] = \mathrm{svd}(C_s + \frac{\mathbf{L}_2}{\beta}) \tag{34}$$

Solution to J_t

$$\begin{aligned} &\equiv \underset{J_t}{\mathrm{argmin}}\ \lambda_4\|J_t\|_* + \frac{\beta}{2}\|C_t - J_t\|_F^2 + <\mathbf{L}_3, C_t - J_t> \\ &\equiv \underset{J_t}{\mathrm{argmin}}\ \lambda_4\|J_t\|_* + \frac{\beta}{2}\|J_t - \big(C_t + \frac{\mathbf{L}_3}{\beta}\big)\|_F^2 \end{aligned} \tag{35}$$

$$J_t \equiv U_{J_t}\mathscr{S}_{\frac{\lambda_4}{\beta}}(\Sigma_{J_t})V_{J_t}^T,\ \text{where}[U_{J_t}, \Sigma_{J_t}, V_{J_t}^T] = \mathrm{svd}(C_t + \frac{\mathbf{L}_3}{\beta}) \tag{36}$$

The pseudo-code with few MATLAB script of our implementation is provided in **Algorithm (1)**. This divide and conquer approach works well for most of the available benchmark dataset, however, to make our method more robust to a real-world setting, we took our idea a step further. We know that high-dimensional data representation can be inferior in the presence of noise and outliers unless some filtering techniques are employed. Therefore, we must introduce the notion of low-dimensional representation on Grassmann manifold which preserves the important geometrical information and lets us get rid of noisy information in the data. Inquest of implementing this idea, we proposed a geometry aware extension to our previously proposed dense NRSfM algorithm.

6 Geometry Aware Idea

6.1 Motivation

The key insight in the last algorithm is; even though the overall complexity of the deforming shape is high, each local deformation may be less complex. Using this idea, we developed a union of local linear subspace approach to solve dense NRSfM problem.

Despite its excellent performance, it has some practical concerns. *Firstly*, the intrinsic issues associated with the modeling of a non-rigidly deforming surface via a **high-dimensional** Grassmannian representation. Now, such a representation may help reconstruct complex 3D deformation but can lead to wrong clustering —curse of dimensionality [57], and it's very important in a joint reconstruction and clustering framework to have suitable clusters of subspaces, else reconstruction may suffer. *Secondly*, the approach to represent local non-linear deformation completely ignores 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 can result in *irredeemable* discontinuity of the trajectories (see Fig.(3)). Hence, temporal representation of the set of shapes using Grassmannians seems not an extremely beneficial choice for modeling dense NRSfM on Grassmannian manifold, **unless prior information is available which in general is not known**. *Lastly*, although the dense NRSfM algorithm proposed in Algorithm (1) works better and faster than the previous methods, it depends on several manual parameters which are inadmissible for practical applications.

Hence, we extend the idea of our previous approach that can overcome the aforementioned limitations with **Algorithm (1)**. The main point we are trying to make is that; reconstruction and grouping of subspace on the same high dimensional Grassmann manifold seem like 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 for grouping Grassmannians [58] [55]. So, inspired from these past work, 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 manifold and cluster subspace on its corresponding low-dimension manifold representation.

We devise an unsupervised approach to efficiently represent the high-dimensional Grassmannians to a lower-dimensional Grassmann manifold via a projection operation. These low-dimensional Grassmannians are represented in such a way, it preserves the local structure of the surface deformation in accordance with its neighboring surfaces. These low-dimensional Grassmannians serves as a potential representative for its high-dimensional Grassmannians for suitable grouping, which subsequently helps improve the reconstruction and representation of the Grassmannians on the high-dimensional Grassmann manifold. Further, we drop the temporal grouping of shapes using Grassmannians to discourage the discontinuity of trajectories (see Fig.(3)).

In essence, our modification is inspired by the previously developed idea and is oriented towards settling its important limitations. Moreover, in contrast to **Algorithm (1)**, we capture the notion of dependent local subspace in a union of subspace algorithm via Grassmannian modeling [14]. The algorithm we proposed is an attempt to supply a more efficient, reliable and practical solution to this problem. Our new formulation gives an efficient framework for modeling dense NRSfM on the Grassmann manifold. We observed empirically that this method is more useful and is as accurate and efficient than **Algorithm (1)**. The performance of this algorithm stands superior in handling noise. The main highlights of this algorithm are as follows:

1) An efficient framework for modeling non-rigidly deforming surface that exploits the advantage of Grassmann manifold representation of different dimensions based on its geometry.
2) 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.
3) An iterative solution to the proposed cost function based on ADMM [34], which is simple to implement and provide results as good as **Algorithm(1)** and in addition to that, it helps improve the 3D reconstruction substantially, in the case of noisy trajectories.

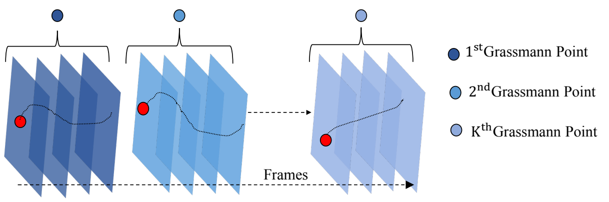


Fig. 3: Temporal representation using Grassmannians in the shape space introduces discontinuity in the overall trajectory of the feature point. Also, to define neighboring subspace dependency graph in the time domain seems very challenging keeping in mind that the activity/expression may repeat. Red circle shows the feature point with its trajectory over frames (Black).

6.2 New Grassmannian Representation

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,

$$\phi_i = \Delta^T \Phi_i \tag{37}$$

Its quite easy to examine that ϕ_i is not a orthogonal matrix and, therefore, may not qualify 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.

$$\Theta_i U_i = \mathbf{qr}(\phi_i) = \Delta^T \Phi_i \tag{38}$$

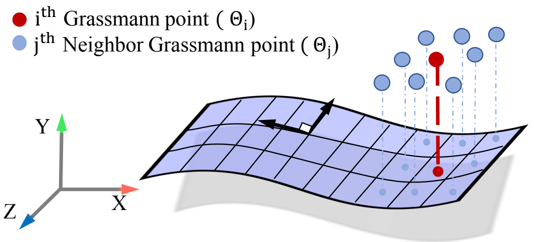
Here, $\mathbf{qr}(.)$ is a function that returns the QR decomposition of the matrix. The $\Theta_i \in \mathbb{R}^{\tilde{d} \times p}$ is an orthogonal matrix and $U_i \in \mathbb{R}^{p \times p}$ is the upper triangular matrix[3]. Using Eq.(38), we represent the equivalence of Φ_i in low dimension as

$$\begin{aligned} \Theta_i &= \Delta^T (\Phi_i U_i^{-1}) \\ \Theta_i &= \Delta^T \Omega_i \end{aligned} \tag{39}$$

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.(4), Fig.(5)), we further constrain our representation as:

$$E(\Delta) = \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Pi(\Theta_i) - \Pi(\Theta_j)\|_2^F \tag{40}$$

[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 ($U_i \in \mathbb{R}^{p \times p}$)



i^{th} Grassmann point (Θ_i)
j^{th} Neighbor Grassmann point (Θ_j)
Y
Z
X

Fig. 4: In contrast to Algorithm (1) representation, the 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 an assumption. However, building such graph in shape space can be tricky.

The parameter '$\mathtt{w_{ij}}$' accommodate the similarity knowledge between the two Grassmannians. Using the **Definition**(3.2) and Eq.(39), we further simplify Eq.(40) as

$$\begin{aligned}
\mathrm{E}(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \mathrm{w_{ij}} \frac{1}{2} \|\Delta^{\mathrm{T}}\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}}\Delta - \Delta^{\mathrm{T}}\Omega_{\mathrm{j}}\Omega_{\mathrm{j}}^{\mathrm{T}}\Delta\|_{\mathrm{F}}^{2} \\
\mathrm{E}(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \mathrm{w_{ij}} \frac{1}{2} \|\Delta^{\mathrm{T}}(\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}} - \Omega_{\mathrm{j}}\Omega_{\mathrm{j}}^{\mathrm{T}})\Delta\|_{\mathrm{F}}^{2} \qquad (41) \\
\mathrm{E}(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \mathrm{w_{ij}} \frac{1}{2} \|\Delta^{\mathrm{T}}(\Lambda_{\mathrm{ij}})\Delta\|_{\mathrm{F}}^{2}
\end{aligned}$$

where, $\Lambda_{\mathrm{ij}} \in \mathrm{Sym}(\mathrm{d})$. The parameter '$\mathrm{w_{ij}}$' (**similarity graph**) is set as $\exp(-d_g^2(\Phi_{\mathrm{i}}, \Phi_{\mathrm{j}}))$ with d_g as the projection metric (see **Definition** (3.2)). Eq.(41) 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.(41). By performing some simple algebraic manipulation, Eq.(41) reduces to

$$\mathbf{Tr}\Big(\Delta^{\mathrm{T}}\big(\sum_{\mathrm{i}=1}^{\mathrm{K}} \lambda_{\mathrm{ii}}\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}}\big)\Delta\Big) \qquad (42)$$

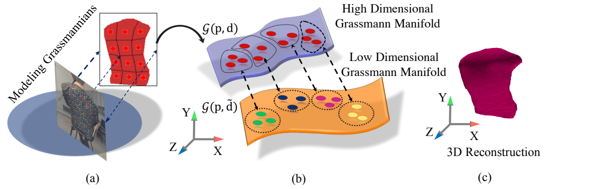
where, $\lambda_{\mathrm{ii}} = \sum_{\mathrm{j}=1}^{\mathrm{K}} \mathrm{w_{ij}}$. Constraining the value of Eq.(42) to 1 provides the overall optimization for an efficient representation of the local non-rigid surface on the Grassmann manifold.

$$\begin{aligned}
&\mathrm{E}(\Delta) \equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \mathrm{w_{ij}} \frac{1}{2} \|\Delta^{\mathrm{T}}(\Lambda_{\mathrm{ij}})\Delta\|_{\mathrm{F}}^{2} \\
&\text{subject to:} \qquad (43) \\
&\mathbf{Tr}\Big(\Delta^{\mathrm{T}}\big(\sum_{\mathrm{i}=1}^{\mathrm{K}} \lambda_{\mathrm{ii}}\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}}\big)\Delta\Big) = 1
\end{aligned}$$

Its easy to verify that the matrix 'Λ' and '$\big(\sum_{\mathrm{i}=1}^{\mathrm{K}} \lambda_{\mathrm{ii}}\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}}\big)$' are symmetric and positive semi-definite, and therefore, the above optimization can be solved as a generalized eigen value problem.

6.3 Solution to $\mathrm{E}(\Delta)$

$$\begin{aligned}
&\mathrm{E}(\Delta) \equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \mathrm{w_{ij}} \frac{1}{2} \|\Delta^{\mathrm{T}}(\Lambda_{\mathrm{ij}})\Delta\|_{\mathrm{F}}^{2} \\
&\text{subject to:} \qquad (44) \\
&\mathbf{Tr}\Big(\Delta^{\mathrm{T}}\big(\sum_{\mathrm{i}=1}^{\mathrm{K}} \lambda_{\mathrm{ii}}\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}}\big)\Delta\Big) = 1
\end{aligned}$$



Modeling Grassmannians
$\mathcal{G}(\mathrm{p,d})$
High Dimensional Grassmann Manifold
Low Dimensional Grassmann Manifold
$\mathcal{G}(\mathrm{p},\tilde{\mathrm{d}})$
Y
Z
X
3D Reconstruction
(a)
(b)
(c)

Fig. 5: 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 optimization equation proposed for $\mathrm{E}(\Delta)$ is a well-studied optimization form and Riemann Conjugate gradient toolbox can be employed to achieve the solution. Nevertheless, we can also derive augmented lagrangian form to solve the same problem. By letting $\mathrm{X} = \big(\sum_{\mathrm{i}=1}^{\mathrm{K}} \lambda_{\mathrm{ii}}\Omega_{\mathrm{i}}\Omega_{\mathrm{i}}^{\mathrm{T}}\big)$ and expanding the Frobenius norm term, we can re-write the equation as:

$$\begin{aligned}
&\mathrm{E}(\Delta) \equiv \underset{\Delta}{\text{minimize}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \frac{\mathrm{w_{ij}}}{2} \mathbf{Tr}\big(\Delta^{\mathrm{T}}\Lambda_{\mathrm{ij}}\Delta\Delta^{\mathrm{T}}\Lambda_{\mathrm{ij}}\Delta\big) \\
&\mathrm{E}(\Delta) \equiv \underset{\Delta}{\text{minimize}}\ \mathbf{Tr}\Big(\Delta^{\mathrm{T}} \sum_{(\mathrm{i,j})}^{\mathrm{K}} \frac{\mathrm{w_{ij}}}{2} \Lambda_{\mathrm{ij}}\Delta^{t-1}\Delta^{(t-1)\mathrm{T}}\Lambda_{\mathrm{ij}}\Delta\Big) \qquad (45) \\
&\text{subject to:} \\
&\mathbf{Tr}\Big(\Delta^{\mathrm{T}}\mathrm{X}\Delta\Big) = 1
\end{aligned}$$

Here, $t-1$ refers to its known value before the current iteration. Now, by assuming $\mathrm{Y} = \frac{\mathrm{w_{ij}}}{2}\Lambda_{\mathrm{ij}}\Delta^{t-1}\Delta^{(t-1)\mathrm{T}}\Lambda_{\mathrm{ij}}$, the above equation simplifies to standard eigen value decomposition problem *i.e,*

$$\begin{aligned}
&\mathrm{E}(\Delta) \equiv \underset{\Delta}{\text{minimize}}\ \mathbf{Tr}(\Delta^{\mathrm{T}}\mathrm{Y}\Delta) \\
&\text{subject to:} \qquad (46) \\
&\mathbf{Tr}\Big(\Delta^{\mathrm{T}}\mathrm{X}\Delta\Big) = 1
\end{aligned}$$

The equivalent Lagrangian function form is given by

$$\mathrm{Tr}(\Delta^{\mathrm{T}}\mathrm{Y}\Delta) + \lambda\Big(1 - \mathbf{Tr}\big(\Delta^{\mathrm{T}}\mathrm{X}\Delta\big)\Big) \qquad (47)$$

Eq.(47) is in the generalized eigen value problem form. Any standard linear algebra package can be used to solve it.

6.4 Geometry Aware Extension

Similar to the previous algorithm, we introduce the local subspace constraint on the shape, we use the notion of self-expressiveness on the non-linear Grassmann manifold space.

$$\begin{aligned}
&\underset{\mathrm{E,C,S^{\sharp}}}{\text{minimize}}\ \|\mathrm{E}\|_{\mathcal{G}}^{2} + \beta_2\|\mathrm{S}^{\sharp}\|_{*} + \beta_3\|\mathrm{C}\|_{*} \\
&\text{subject to: } \mathrm{S}^{\sharp} = f(\mathrm{S}),\ \mathrm{S} = \mathrm{SC} + \mathrm{E}
\end{aligned} \qquad (48)$$

As defined before $f : \mathrm{S} \in \mathbb{R}^{3\mathrm{F}\times\mathrm{P}} \mapsto \mathrm{S}^{\sharp} \in \mathbb{R}^{3\mathrm{P}\times\mathrm{F}}$ and $\mathrm{C} \in \mathbb{R}^{\mathrm{P}\times\mathrm{P}}$ as the coefficient matrix. We know from our previous discussion that the Grassmann manifold is isometrically equivalent to the symmetric idempotent matrix [56]. So, we embed the Grassmann manifold into symmetric matrix manifold to define the self-expressiveness. Let $\xi_{\mathrm{s}} = \{\Theta_1, \Theta_2, ..., \Theta_{\mathrm{K}}\}$ be the set of Grassmannians on a low dimensional Grassmann manifold. The elements of ξ_{s} are the projection of high dimensional Grassmannian representation of the columns of 'S' matrix. Let $\mathcal{X} = \{(\Theta_1\Theta_1^{\mathrm{T}}), (\Theta_2\Theta_2^{\mathrm{T}}), ..., (\Theta_{\mathrm{K}}\Theta_{\mathrm{K}}^{\mathrm{T}})\}$

be its embedding onto symmetric matrix manifold. Using such embedding techniques we re-write Eq.(48) as

$$\begin{aligned} &\underset{\mathtt{E},\tilde{\mathtt{C}},\mathtt{S}^{\sharp}}{\text{minimize}}\ \|\mathtt{E}\|_{\mathtt{F}}^2+\beta_2\|\mathtt{S}^{\sharp}\|_*+\beta_3\|\tilde{\mathtt{C}}\|_* \\ &\text{subject to:}\mathtt{S}^{\sharp}=f(\mathtt{S}),\mathcal{X}=\mathcal{X}\tilde{\mathtt{C}}+\mathtt{E} \end{aligned} \tag{49}$$

where, $\tilde{\mathtt{C}} \in \mathbb{R}^{\mathtt{K}\times\mathtt{K}}$ and $\mathcal{X} \in \mathbb{R}^{\tilde{\mathtt{d}}\times\tilde{\mathtt{d}}\times\mathtt{K}}$ 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.

Similar to previous notations, let $\mathbf{P} \in \mathbb{R}^{1\times\mathtt{P}}$ be an ordering vector that contains the index of columns of $\mathtt{S}$. Also, using the the function definition form $\{(\text{output}, \text{function}(.)) : \text{definition}\}$, we define f_h as

$$\begin{aligned} &\big\{\big(\tilde{\xi}_{\mathtt{s}}, f_h(\Delta,\xi_{\mathtt{s}})\big) : \tilde{\xi}_{\mathtt{s}} = \{\Theta_{\mathtt{i}}\}_{\mathtt{i}=1}^{\mathtt{K}}, \Theta_{\mathtt{i}} = \Delta^{\mathtt{T}}(\Phi_{\mathtt{i}}\mathtt{U}_{\mathtt{i}}^{-1}), \\ &\text{where, } \Delta = \text{solution to the minimization of Eq.(43)}\big\} \end{aligned} \tag{50}$$

Intuitively, The function (f_h) projects the Grassmannians to a lower dimension in accordance with the neighbors using Eq.(43)

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

$$\begin{aligned} &\underset{\mathtt{E},\tilde{\mathtt{C}},\mathtt{S},\mathtt{S}^{\sharp}}{\text{minimize}}\frac{1}{2}\|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2+\beta_1\|\mathtt{E}\|_{\mathtt{F}}^2+\beta_2\|\mathtt{S}^{\sharp}\|_*+\beta_3\|\tilde{\mathtt{C}}\|_* \\ &\text{subject to:} \\ &\mathtt{S}^{\sharp}=f(\mathtt{S}),\mathcal{X}=\mathcal{X}\tilde{\mathtt{C}}+\mathtt{E}, \\ &\xi_{\mathtt{s}}=f_g(\mathbf{P},\mathtt{S},\mathtt{K},\mathtt{p}),\tilde{\xi}_{\mathtt{s}}=f_h(\Delta,\xi_{\mathtt{s}}), \\ &\mathtt{S}=f_s(\xi_{\mathtt{s}},\Sigma,\xi_{\mathtt{v}},\mathtt{K},\mathtt{p}),\mathbf{P}=f_p(\tilde{\xi}_{\mathtt{s}},\tilde{\mathtt{C}},\mathbf{P}_{\mathtt{so}}) \end{aligned} \tag{51}$$

where $\mathbf{P}_{\circ}$ vector contains the initial ordering of the columns of '$\mathtt{W}$' and '$\mathtt{S}$'. The function (f_p) provides the ordering index to rearrange the columns of '$\mathtt{S}$' matrix to be consistent with '$\mathtt{W}$' matrix. This is important because, grouping the set of columns of '$\mathtt{S}$' over iteration, disturbs its initial arrangements. The definition of f_g, f_s and f_p is same as outlined in Eq(19), Eq(20) and Eq(21) respectively.

6.5 Solution

The optimization proposed in Eq.(51) is a coupled optimization problem. Several methods of Bi-level optimization can be used to solve such minimization problem [59], [60]. Nevertheless, we propose ADMM [34] 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.(43), 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 $\mathcal{X}$, we introduce auxiliary variable $\tilde{\mathtt{C}} = \mathtt{Z}$. We apply these operations to our optimization problem to get the following Augmented Lagrangian form:

$$\begin{aligned} &\underset{\mathtt{Z},\tilde{\mathtt{C}},\mathtt{S},\mathtt{S}^{\sharp}}{\text{minimize}}\frac{1}{2}\|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}^2+\beta_1\|\mathcal{X}-\mathcal{X}\tilde{\mathtt{C}}\|_{\mathtt{F}}^2+\beta_2\|\mathtt{S}^{\sharp}\|_*+\frac{\beta}{2}\|\mathtt{S}^{\sharp}-f(\mathtt{S})\|_{\mathtt{F}}^2+ \\ &\qquad <\mathbf{L_1},\mathtt{S}^{\sharp}-f(\mathtt{S})>+\beta_3\|\mathtt{Z}\|_*+\frac{\beta}{2}\|\tilde{\mathtt{C}}-\mathtt{Z}\|_{\mathtt{F}}^2+<\mathbf{L_2},\tilde{\mathtt{C}}-\mathtt{Z}> \\ &\text{subject to:} \\ &\xi_{\mathtt{s}}=f_g(\mathbf{P},\mathtt{S},\mathtt{K},\mathtt{p}),\ \tilde{\xi}_{\mathtt{s}}=f_h(\Delta,\xi_{\mathtt{s}}) \\ &\mathtt{S}=f_s(\xi_{\mathtt{s}},\Sigma,\xi_{\mathtt{v}},\mathtt{K},\mathtt{p}),\ \mathbf{P}=f_p(\tilde{\xi}_{\mathtt{s}},\tilde{\mathtt{C}},\mathbf{P}_{\circ}) \end{aligned} \tag{52}$$

Algorithm 2 Geometry Aware Dense NRSfM

```
Require: $\mathtt{W}, \mathtt{R}, \{\beta_{\mathtt{i}}\}_{\mathtt{i}=1}^3$, $\beta$=$e^{-2}$, $\beta_m$=$e^8$, $\varepsilon$=$e^{-10}$, $c$ =1.1, $\mathtt{K}$;
Initialize: $\mathtt{S}$=pinv($\mathtt{R}$)$\mathtt{W}$, $\mathtt{S}^{\sharp}$=$f(\mathtt{S})$, $\mathtt{Z}$=$\mathbf{0}$, $\{\mathbf{L}_{\mathtt{i}}\}_{\mathtt{i}=1}^2$=$\mathbf{0}$, $\tilde{\mathtt{d}}$;
        $\Delta$ = [$\mathbf{I}_{\tilde{\mathtt{d}}\times\tilde{\mathtt{d}}}$; random values], $\mathtt{p}$ %top singular values
        $\mathbf{P}_{\mathtt{so}}$ = kmeans++($\mathtt{S},\mathtt{K}$), iter = 1, $\mathbf{P}_{\text{store}}(\text{iter},:)$ = $\mathbf{P}_{\mathtt{so}}$,
        $\mathbf{P}$ = $\mathbf{P}_{\mathtt{so}}$
  Define: $\mathscr{S}_{\tau}(\mathtt{x}) := @(\mathtt{x},\tau)\text{sign}(\mathtt{x}).*\max(\text{abs}(\mathtt{x})-\tau,0)$;
  while not converged do
     1. $\mathtt{S} := \text{mldivide}\big(\mathtt{R}^{\mathtt{T}}\mathtt{R}+\beta\mathtt{I},\ \beta(f^{-1}(\mathtt{S}^{\sharp})+\frac{f^{-1}(\mathbf{L_1})}{\beta})+\mathtt{R}^{\mathtt{T}}\mathtt{W}\big)$;
     2. $\xi_{\mathtt{s}} := f_g(\mathbf{P},\mathtt{S},\mathtt{K},\mathtt{p})$; see Eq.(19)
     3. $\mathtt{W} :=$ arrange_column($\mathbf{P},\mathtt{W}$)
     4. Update the similarity matrix '$\mathtt{w}_{\mathtt{ij}}$' using $\xi_{\mathtt{s}}$. §6.2
     5. $\tilde{\xi}_{\mathtt{s}} := f_h(\xi_{\mathtt{s}},\Delta); \text{s.t}, \Delta \equiv \underset{\Delta}{\text{minimize}}\ \mathtt{E}(\Delta)$; see Eq.(50)
     6. $\Gamma_{\mathtt{ij}} = \mathbf{Tr}[(\Theta_{\mathtt{j}}^{\mathtt{T}}\Theta_{\mathtt{i}})(\Theta_{\mathtt{i}}^{\mathtt{T}}\Theta_{\mathtt{j}})]; \Gamma = (\Gamma_{\mathtt{ij}})_{\mathtt{ij}=1}^{\mathtt{K}}; \mathtt{L} = \mathbf{Chol}(\Gamma)$
     7. $\tilde{\mathtt{C}} := \big(2\beta_1\mathtt{LL}^{\mathtt{T}}+\beta(\mathtt{Z}-\frac{\mathbf{L_2}}{\beta})\big)\ \big(2\beta_1\mathtt{LL}^{\mathtt{T}}+\beta\mathtt{I}\big)^{-1}$;
     8. $\mathbf{P} := f_p(\tilde{\xi}_{\mathtt{s}},\tilde{\mathtt{C}},\mathbf{P})$;
     9. $\mathtt{S} := f_s(\xi_{\mathtt{s}},\Sigma,\xi_v,\mathtt{K},\mathtt{p})$; see Eq.(19), Eq.(20)
     10. $\mathtt{S}^{\sharp} := \mathtt{U}_{\mathtt{s}}\mathscr{S}_{\frac{\beta_2}{\beta}}(\Sigma_{\mathtt{s}})\mathtt{V}_{\mathtt{s}}; \text{s.t}, [\mathtt{U}_{\mathtt{s}},\Sigma_{\mathtt{s}},\mathtt{V}_{\mathtt{s}}] := \text{svd}(f(\mathtt{S})-\frac{\mathbf{L_1}}{\beta})$
     11. $\mathtt{Z} := \mathtt{U}_{\mathtt{z}}\mathscr{S}_{\frac{\beta_3}{\beta}}(\Sigma_{\mathtt{z}})\mathtt{V}_{\mathtt{z}}; \text{s.t}, [\mathtt{U}_{\mathtt{z}},\Sigma_{\mathtt{z}},\mathtt{V}_{\mathtt{z}}] := \text{svd}(\tilde{\mathtt{C}}+\frac{\mathbf{L_2}}{\beta})$;
     12. $\mathbf{L_1} := \mathbf{L_1}+\beta(\mathtt{S}^{\sharp}-f(\mathtt{S})); \mathbf{L_2} := \mathbf{L_2}+\beta(\tilde{\mathtt{C}}-\mathtt{Z})$
     13. iter := iter + 1; $\mathbf{P}_{\text{store}}(\text{iter},:) := \mathbf{P}$;
     14. $\beta := \min(\beta_m, c\beta)$;
     15. gap :=   $\max\{\|\mathtt{S}^{\sharp}-f(\mathtt{S})\|_\infty, \|\tilde{\mathtt{C}}-\mathtt{Z}\|_\infty\}$;
     $(\text{gap} < \varepsilon)\vee(\beta > \beta_m) \rightarrow$ break; %convergence check
  end while
  return $\mathtt{S}$;
$e_{\mathtt{3D}}$ = Estimate_error ($\mathtt{S}_{\text{est}} = \mathtt{S}, \mathtt{S}_{\mathtt{GT}}, \mathbf{P}_{\text{store}}$); %use Eq.(53)
```

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.(43) and Eq.(20) respectively. More generally, the solution to the optimization in Eq.(43) is obtained by solving it as a generalized eigenvalue problem. To keep the order of columns of '$\mathtt{S}$' matrix consistent with '$\mathtt{W}$' matrix $f_p(.)$ provides the ordering index. We provide the implementation details of our method with suitable MATLAB commands in the **Algorithm (2)**.

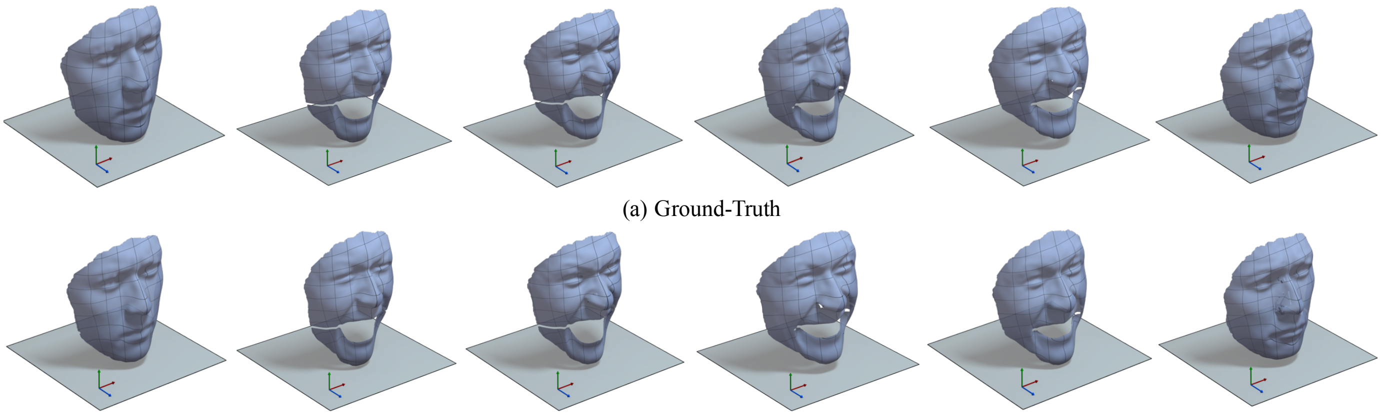
7 Experimental Evaluation

We performed extensive qualitative and quantitative evaluations on the available standard benchmark datasets [3] [35] [30]. To keep our evaluations consistent with the previous methods, we compute the average 3D reconstruction quality of the estimated shape '$\mathtt{S}_{\text{est}}$' using the following equation

$$\mathtt{e}_{\mathtt{3D}} = \frac{1}{\mathtt{F}}\sum_{\mathtt{i}=1}^{\mathtt{F}}\frac{\|\mathtt{S}_{\text{est}}^{\mathtt{i}}-\mathtt{S}_{\mathtt{GT}}^{\mathtt{i}}\|_{\mathtt{F}}}{\|\mathtt{S}_{\mathtt{GT}}^{\mathtt{i}}\|_{\mathtt{F}}} \tag{53}$$

here, '$\mathtt{S}_{\mathtt{GT}}$' denotes the ground-truth 3D shape matrix. The qualitative results for both the algorithm looks very similar, yet, they are statistically different (see Table 1).

Initialization: The initialization for both the algorithms are straight-forward and outlined in the respective algorithm table. In brief, we used Intersection method [5] to estimate the rotation matrix and initialize $\mathtt{S} = \mathbf{pinv}(\mathtt{R})\mathtt{W}$. The initial grouping of the trajectories or columns of $\mathtt{S}$ is done using k-means++ algorithm [31]. These initial grouping is used to initialize the ordering vector $\mathbf{P}_{\circ}$,



(a) Ground-Truth

(b) Our Result

Fig. 6: Reconstruction results obtained on synthetic dense face dataset (face sequence 4). **Top row** : Ground-truth 3D points, **Bottom row** : Recovered 3D shape using Algorithm 1. Visually, the 3D reconstruction results recovered using both the algorithms looks very similar.

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.(44) to initialize $\tilde{\xi}$ and store corresponding singular values. The similarity matrix or graph in Eq.(44) is constructed using the distance measure between the Grassmannians in the embedding space §6.2.

1. Results on synthetic Face dataset: The synthetic face dataset is composed of four distinct sequence [3] 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 ±30° and ±90° respectively. Sequence 3 and Sequence 4 are 99 frame long video that contains high frequencies and low frequencies rotation respectively. It's a challenging dataset mainly due to different rotation frequencies and deformations in each of the facial expression sequence. Table (1) shows the statistical results obtained on these four sequences using both of our algorithms. Fig.(6) show the qualitative results on face sequence 4 of the dataset.

2. Results on Paper and T-shirt dataset: To evaluate our performance on smooth deforming surfaces, we used Varol et.al. [35] 'kinect_paper' and 'kinect_tshirt' datasets. This dataset provides real condition to test the performance of NRSfM algorithm. It provides sparse SIFT [61] 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. algorithm [62] to estimate the measurement matrix. Numerically, we compute the correspondence of the deforming subject within $\mathtt{x}_{\mathtt{w}} = (253, 253, 508, 508)$, $\mathtt{y}_{\mathtt{w}} = (132, 363, 363, 132)$ 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. Fig.(7) shows couple of 3D reconstruction results on these sequence with comparative results specified in Table (1).

3. Results on Actor dataset: Beeler et.al. [30] 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 [16]. Fig.(7) show the dense detailed reconstruction that is achieved using our algorithms. Table (1) clearly indicates the superior performance of our approach on this high-quality dense dataset.

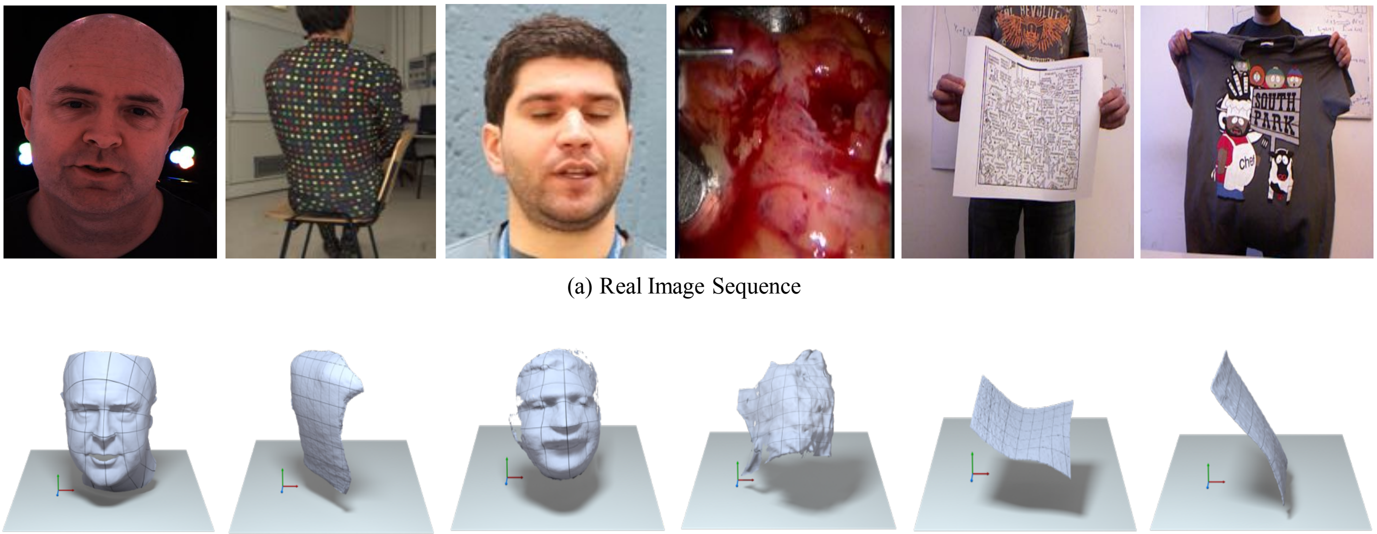
4. Results on Face, Heart, Back dataset: To evaluate the variational approach to dense NRSfM Garg et.al. [3] introduced this dataset. This dataset is composed of monocular video's captured in a natural environment with varying lighting condition and large displacements. It consists of three different videos with 120, 150 and 80 frames for face sequence, back sequence and heart sequence respectively. Furthermore, this dataset provides dense 2D feature tracks for these 3 categories. Specifically, face, back and heart sequence is composed of 28332, 20561, and 68295 features tracks respectively. Ground-truth 3D is not available with this dataset for evaluation. Fig.(7) show some qualitative 3D reconstruction results on face, back and heart sequence. The qualitative results —shown in Fig.(7), demonstrate that our approach is able to estimate the 3D reconstruction of a deforming subject reliably and accurately.

7.1 Algorithmic Analysis

We performed several other experiments to understand the behavior of our algorithm under different input parameters and evaluation setup. In practice these experiments help analyze the practical applicability of our algorithm.

1. Processing Time and Convergence: The execution time for both the algorithm (Algorithm 1 and Algorithm 2) is more or less same. Nevertheless, in comparison to other previous methods our processing time is far better. We computed the processing time on commodity desktop machine with 16GB RAM suing MATLAB R2019b software. Fig.(8(a)) show the processing time of our method in comparison to other methods on different datasets. Ideally, our algorithm takes 120-150 iteration to provide an optimal solution to the problem.

2. Performance over noisy trajectories: We utilized the standard experimental procedure to analyze the behavior of our algorithm



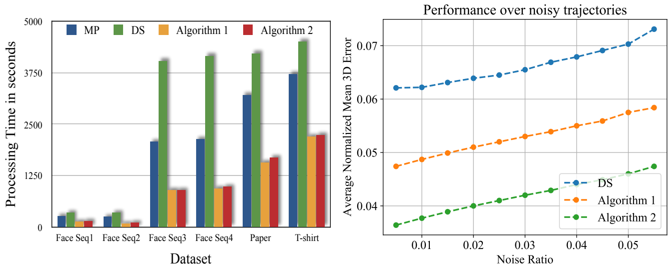
(a) Real Image Sequence

(b) 3D Reconstruction Results

Fig. 7: Reconstruction results obtained on real dataset sequence. **Top row**: Real image sequence, **Bottom row**: Recovered 3D shape using our approach. **Left to Right**: Actor [30], Back [3], Face [3], Heart [3], kinect_paper [35], kinect_tshirt [35] dataset.

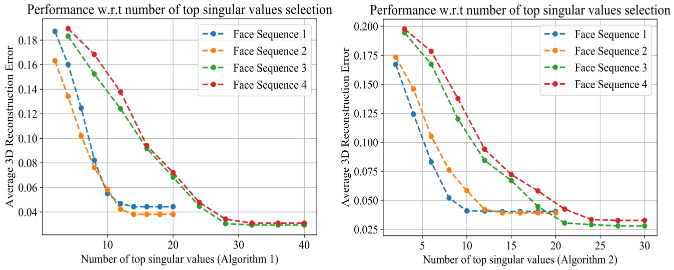
Dataset ↓ / Method →	MP	PTA	CSF1	CSF2	DV	DS	SMSR	Algorithm 1	Algorithm 2
Face Sequence 1	0.0926	0.1559	0.5325	0.4677	0.0531	0.0636	0.1893	0.0443	**0.0404**
Face Sequence 2	0.0819	0.1503	0.9266	0.7909	0.0457	0.0569	0.2133	**0.0381**	0.0392
Face Sequence 3	0.1057	0.1252	0.5274	0.5474	0.0346	0.0374	0.1345	0.0294	**0.0280**
Face Sequence 4	0.0717	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 [16] and DV [3] 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)

Fig. 8: (a) Processing Time Comparison (b) Performance Comparison on Noisy Trajectories

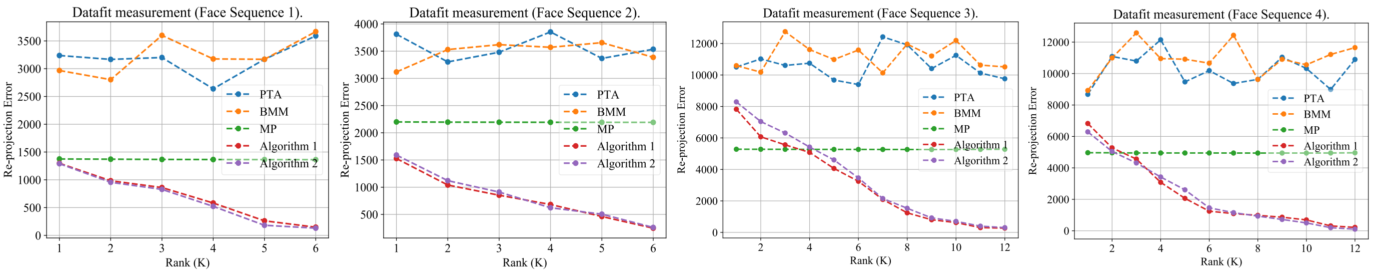


(a) (b)

Fig. 9: Change in the average 3D reconstruction accuracy with respect to number of singular vectors and singular values used.

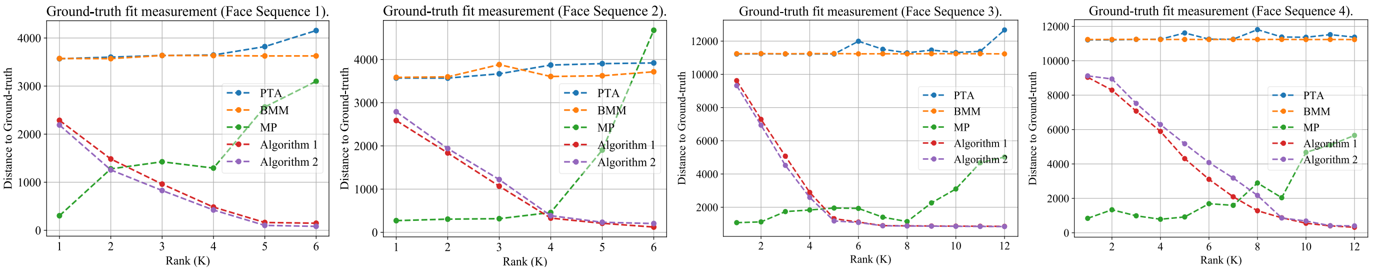
under different noise levels. Similar to the work of Lee et.al. [11], we added the Gaussian noise to the input trajectories. The standard deviation of the noise are adjusted as $\sigma_g = \lambda_g \max\{|\mathtt{W}|\}$ with λ_g varying from 0.01 to 0.055. Fig.(8(b)) show the quantitative comparison of our approach with recent algorithm DS [17]. The graph show the average 3D reconstruction error of all the four synthetic face dataset [3]. The statistics indicate that our algorithms are more resilient to noise than other competing methods. Specifically, Algorithm 2 performs better with noisy data due to the low-dimensional projection of the Grassmannians to perform grouping, which inherently provides robust representation of subspace in presence of noisy trajectories.

3. Performance with change in the number of singular values: The integral value of ‘p’ in $\mathscr{G}(\mathtt{p},\mathtt{d})$ i.e., the number of top singular vectors to represent Grassmannians and, corresponding singular values to perform reconstruction can directly affect the performance of our algorithm. Yet, it has been observed over several experiments that we need relatively few singular values and singular vectors —in comparison to the number of trajectories, to recover dense 3D reconstruction of the deforming object. Fig.(9(a)) and Fig. (9(b)) show the change in average 3D reconstruction with the different values of ‘p’ for synthetic face dataset [3]. The graph shows that for both Algorithm 1 and Algorithm 2, 10-15 singular values are good enough for Face Sequence 1-2 and, 25-30 singular values are sufficient for Face Sequence 3-4. Note that these sequence have 28,880 trajectories and therefore, to reconstruct each vectorial point can be severely expensive. In constrast, our linear subspace representation can handle it easily.



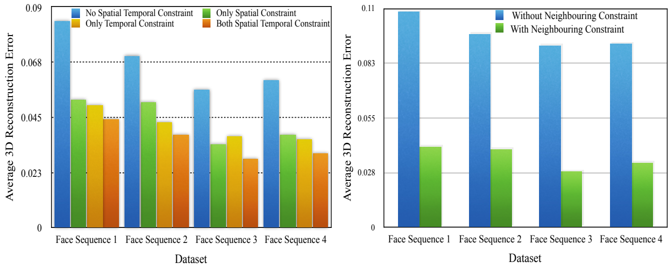
(a) (b) (c) (d)

Fig. 10: The measure of re-projection error constraint i.e $\|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}$ as a function of rank on synthetic face dataset [3].



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

Fig. 11: The measure of ground-truth shape fit i.e, the distance between the obtained shape and ground truth shape ($\|\mathtt{S}_{\mathtt{GT}}-\mathtt{S}\|_{\mathtt{F}}$) as a function of rank on synthetic face dataset [3].



(a) (b)

Fig. 12: (a)-(b) Ablation test results for Algorithm 1 and Algorithm 2 respectively.

4. Analysis of Data-fit with Variation in the Rank: Inspired by the recent work [63], we performed this test to analyse the dependence of the algorithm on the output rank. Specifically, we measure the datafit and ground-truth fit which is defined as $\|\mathtt{W}-\mathtt{RS}\|_{\mathtt{F}}$ and $\|\mathtt{S}_{\mathtt{GT}}-\mathtt{S}\|_{\mathtt{F}}$ respectively. This experiment revels the competence of the algorithm to fit the trajectory/shape based on the selection of the rank. The Fig.(10) and Fig.(11) clearly indicates that sparse NRSfM algorithm [5] [2] fails to handle the dense deforming shapes. Neither, reprojection error nor ground-truth fit is maintained with increase in the 'K' value (rank). Additionally, dense NRSfM approach with only global constraint [9] (MP) fails to capture the local deformation properly, hence, datafit to the ground-truth fails to correlate with the re-projection error with increase in the rank (K value). In contrast, our algorithm has expected trend to both reprojection error and ground-truth error with the variation in the rank. Both of our algorithms have high correlation between the two measures. Note that MP [9] [64] algorithm estimates both rotation and translation for NRSfM problem, however, for consistency we did not consider the translation component for plotting these graphs.

5. Ablation Test: In this paper, we proposed two optimization algorithms that are composed of several constraints. To understand the importance of each constraint, we performed an ablation test. Algorithm 1 has both spatial and temporal subspace constraint, whereas, Algorithm 2 has spatial subspace constraint along with spatial neighbouring constraint. To perform this task for Algorithm 1, we observe the performance of our formulation under four different setups: (a) without any spatio-temporal constraint (b) with only spatial constraint (c) with only temporal constraint (TP), and (d) with both the spatial-temporal constraint. Fig.(12(a)) show the variations in average 3D reconstruction errors under these setups on four synthetic face sequence. The statistics clearly illustrate the importance of both the constraints in our formulation. Similarly, for Algorithm 2, we analysed the performance before and after imposing the neighboring constraint. The result is shown in Fig.(12(b)) that demonstrates the importance of the imposed constraint. Just to remind the readers that for Algorithm 2, we did not consider the temporal constraint. We argued that temporal information are generally not available in real-world cases.

6. Dependence of the algorithm 2 on variable $\tilde{\mathtt{d}}$: Dimensionality reduction to group the grassmann points is one of the critical aspect of Algorithm 2. To determine the dimension to which we should project for better results is a key-concern. We used well-known procedure of cumulative energy of singular vectors to get the value of $\tilde{\mathtt{d}}$. Mathematically, let Ω be the set that stack all the Grassmannians and $\sigma_{\mathtt{i}}$ be the $\mathtt{i}^{\mathrm{th}}$ singular value of $\Omega\Omega^{\mathtt{T}}$, then

$$\tilde{\mathtt{d}} = \underset{\mathtt{d}_{\mathrm{opt}}}{\operatorname{argmin}} \frac{\sum_{\mathtt{i}=1}^{\mathtt{d}_{\mathrm{opt}}} \sigma_{\mathtt{i}}}{\sum_{\mathtt{i}=1}^{\mathtt{d}} \sigma_{\mathtt{i}}} \geq \tau \tag{54}$$

where τ can vary from 0 to 1 and $\mathtt{d}_{\mathrm{opt}}$ (optimal dimension) is a positive integer. We put $\tau = 0.97$ for all our experiment. Fig.(13) show the variations in the reconstruction error with the value of τ for synthetic face dataset [3]. It is observed that for different dataset the value of suitable $\tilde{\mathtt{d}}$ is different. The point to note is that if the reduced dimension is less than the intrinsic dimension, the samples may lose important information for better grouping of

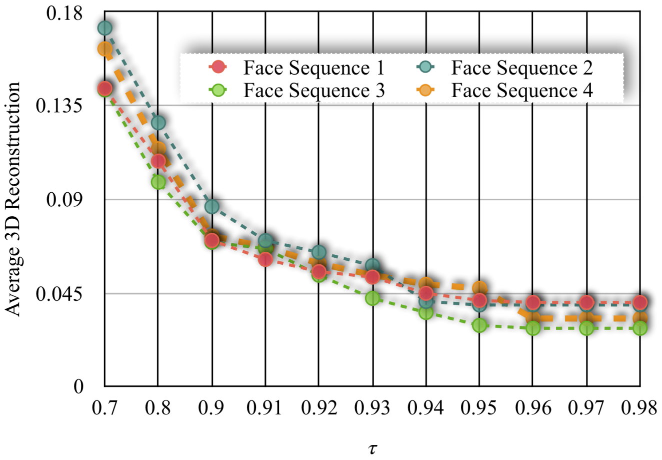


Fig. 13: Variation in the 3D reconstruction accuracy w.r.t τ.

Grassmannians. For our task, in general, $\tau = 0.97$ works well for all the dataset.

8 Practical Limitations

Firstly, our method assumes fairly good dense 2D feature correspondence is provided as input. Nonetheless, estimating robust dense 2D feature correspondences for a deforming surface across frames in itself is a very challenging problem to solve. The main challenges come from the fact that the illumination of the deforming object keeps changing over time. Consequently, a passive approach to establish correspondences may lead to wrong results. Secondly, our representation may fail to handle non-rigid deformation such as stretching and squashing of an object. For example: stretching a rubber sheet or deflating a balloon. Such deformations are hard to handle due to substantial change in the global structure of the shape —area/volume or projected object size can differ considerably over frames. Lastly, object deformation recorded under a distinct camera trajectory can provide different results.

9 Conclusion and Future Work

In this work, we introduced a new representation to solve the problem of dense non-rigid structure from motion. Exploiting both local and global deformation constraints, our algorithm uses the new representation to make the idea of joint segmentation and reconstruction scalable and therefore, is able to obtain the 3D structure of dense deforming surfaces with higher accuracy. Employing a unified spatial-temporal idea to blend the information from both shape and trajectory space, our algorithm demonstrated leading performance on the benchmark datasets. Later, we extended our formulation to a more practical setting, where, temporal shape information is not known a prior and input can be noisy. We used the assumption that a group of neighboring trajectories may span a similar linear subspace. To make our formulation robust to noise, we project the manifold representation to lower dimension for better clustering, thereby implicitly improving the 3D reconstruction. In particular, our algorithm shows notable accuracy in the presence of noise and complex deformations, where other methods may fail.

It has been observed that when the same object deformation is recorded under different camera motion, the non-rigid structure from motion algorithms behaves differently. In the future, we plan to extend our algorithm to handle such a situation robustly. Additionally, how far can we apply the new finding on smooth motion assumption in our pipeline and, when the low-rank model may fail is left as an extension to this work [40].

References

[1] E. Jain, Y. Sheikh, M. Mahler, and J. Hodgins, "Augmenting hand animation with three-dimensional secondary motion," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 2010, pp. 93–102.

[2] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Nonrigid structure from motion in trajectory space," in *Advances in neural information processing systems*, 2009, pp. 41–48.

[3] R. Garg, A. Roussos, and L. Agapito, "Dense variational reconstruction of non-rigid surfaces from monocular video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1272–1279.

[4] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2000, pp. 690–696.

[5] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 101–122, 2014.

[6] P. F. Gotardo and A. M. Martinez, "Non-rigid structure from motion with complementary rank-3 spaces," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3065–3072.

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

[8] P. F. Gotardo and A. M. Martinez, "Kernel non-rigid structure from motion," in *IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 802–809.

[9] M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stovsic, and M. Dodig, "Optimal metric projections for deformable and articulated structure-from-motion," *International journal of computer vision*, vol. 96, no. 2, pp. 252–276, 2012.

[10] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1442–1456, 2011.

[11] M. Lee, J. Cho, C.-H. Choi, and S. Oh, "Procrustean normal distribution for non-rigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1280–1287.

[12] M. Lee, J. Cho, and S. Oh, "Consensus of non-rigid reconstructions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4670–4678.

[13] S. Kumar, Y. Dai, and H. Li, "Multi-body non-rigid structure-from-motion," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 148–156.

[14] V. Larsson and C. Olsson, "Compact matrix factorization with dependent subspaces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017*, vol. 2017. Institute of Electrical and Electronics Engineers Inc., 2017, pp. 4361–4370.

[15] S. Kumar, "A simple prior-free method for non-rigid structure-from-motion factorization : Revisited," *CoRR*, vol. abs/1902.10274, 2019.

[16] M. D. Ansari, V. Golyanik, and D. Stricker, "Scalable dense monocular surface reconstruction," *arXiv preprint arXiv:1710.06130*, 2017.

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

[18] D. Sun, X. Yang, M.-Y. Liu, and J. 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*, 2018, pp. 8934–8943.

[19] J. Hur and S. Roth, "Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 312–321.

[20] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey, "Complex non-rigid motion 3d reconstruction by union of subspaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1542–1549.

[21] S. Kumar, Y. Dai, and H. Li, "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion," *Pattern Recognition*, vol. 71, pp. 428–443, May 2017.

[22] S. H. N. Jensen, A. Del Bue, M. E. B. Doest, and H. Aanæs, "A benchmark and evaluation of non-rigid structure from motion," *arXiv preprint arXiv:1801.08388*, 2018.

[23] R. Garg, A. Roussos, and L. Agapito, "A variational approach to video registration with subspace constraints," *International journal of computer vision*, vol. 104, no. 3, pp. 286–314, 2013.

[24] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematicae*, vol. 80, no. 2, pp. 199–220, 2004.

[25] P. Dollár, V. Rabaud, and S. Belongie, "Non-isometric manifold learning: Analysis and an algorithm," in *International Conference on Machine Learning*, 2007, pp. 241–248.

[26] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition,*. IEEE, 2009, pp. 2790–2797.

[27] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

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

[30] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, "High-quality passive facial performance capture using anchor frames," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 75.

[31] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[32] R. Yu, C. Russell, N. D. Campbell, and L. Agapito, "Direct, dense, and deformable: template-based non-rigid 3d reconstruction from rgb video," in *IEEE International Conference on Computer Vision*, 2015, pp. 918–926.

[33] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," in *ACM Transactions on Graphics*, vol. 28, no. 5, 2009, p. 175.

[34] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[35] A. Varol, M. Salzmann, P. Fua, and R. Urtasun, "A constrained latent variable model," in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2012, pp. 2248–2255.

[36] S. Kumar, A. Cherian, Y. Dai, and H. Li, "Scalable dense non-rigid structure-from-motion: A grassmannian perspective," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 254–263.

[37] S. Kumar, "Jumping manifolds: Geometry aware dense non-rigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5346–5355.

[38] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[39] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3d reconstruction of a moving point from a series of 2d projections," in *European conference on computer vision*. Springer, 2010, pp. 158–171.

[40] S. Kumar, "Non-rigid structure from motion: Prior-free factorization method revisited," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 51–60.

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

[42] C. Russell, J. Fayad, and L. Agapito, "Dense non-rigid structure from motion," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012, pp. 509–516.

[43] S. Kumar, Y. Dai, and H. Li, "Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4649–4657.

[44] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4058–4066.

[45] M. Gallardo, T. Collins, A. Bartoli, and F. Mathias, "Dense non-rigid structure-from-motion and shading with unknown albedos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3884–3892.

[46] M. Marquez and J. Costeira, "Optimal shape from motion estimation with missing and degenerate data," in *IEEE Workshop on Application of Computer Vision (WACV)*, 2008.

[47] V. Golyanik, A. Jonas, D. Stricker, and C. Theobalt, "Intrinsic dynamic shape prior for fast, sequential and dense non-rigid structure from motion with detection of temporally-disjoint rigidity," *arXiv preprint arXiv:1909.02468*, 2019.

[48] S. Hauberg, A. Feragen, and M. J. Black, "Grassmann averages for scalable robust pca," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3810–3817.

[49] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *International conference on Machine learning*. ACM, 2008, pp. 376–383.

[50] M. Harandi, C. Sanderson, C. Shen, and B. Lovell, "Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3120–3127.

[51] H. E. Cetingul and R. Vidal, "Intrinsic mean shift for clustering on stiefel and grassmann manifolds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1896–1902.

[52] A. Pasko and V. Adzhiev, "Function-based shape modeling: mathematical framework and specialized language," in *International Workshop on Automated Deduction in Geometry*. Springer, 2002, pp. 132–160.

[53] Y. Sheng, P. Willis, G. G. Castro, and H. Ugail, "Facial geometry parameterisation based on partial differential equations," *Mathematical and Computer Modelling*, vol. 54, no. 5, pp. 1536–1548, 2011.

[54] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Low rank representation on grassmann manifolds: An extrinsic perspective," *arXiv preprint arXiv:1504.01807*, 2015.

[55] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices," in *European conference on computer vision*. Springer, 2014, pp. 17–32.

[56] Y. Chikuse, *Statistics on special manifolds*. Springer Science & Business Media, 2012, vol. 174.

[57] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[58] Z. Huang, R. Wang, S. Shan, and X. 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*, 2015, pp. 140–149.

[59] J. F. Bard, *Practical bilevel optimization: algorithms and applications*. Springer Science & Business Media, 2013, vol. 30.

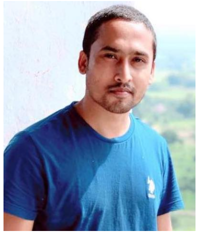
[60] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo, "On differentiating parameterized argmin and argmax problems with application to bi-level optimization," *arXiv preprint arXiv:1607.05447*, 2016.

[61] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer vision*, 1999, pp. 1150–1157.

[62] R. Garg, A. Roussos, and L. Agapito, "Robust trajectory-space tv-l1 optical flow for non-rigid sequences," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 2011, pp. 300–314.

[63] M. V. Örnhag, C. Olsson, and A. Heyden, "A unified optimization framework for low-rank inducing penalties," *arXiv preprint arXiv:2001.08415*, 2020.

[64] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2898–2905.



Suryansh Kumar presently holds the position of Professorship for Computer Vision in the Department of Information Technology and Electrical Engineering at ETH Zürich, where he is advised by Prof. Dr. Luc Van Gool on 3D vision projects. He received Ph.D. in Engineering and Computer Science from the Australian National University in 2019. His Ph.D. dissertation on "Non-Rigid Structure from Motion" is nominated for J.G. Crawford Prize 2019 at the Australian National University. He received M.S in Computer Science and Engineering from International Institute of Information Technology, Hyderabad (IIIT-H) in 2013. Before joining Australian National University, he worked as a Visiting Scientist in the e-Motion Group at INRIA Rhône Alpes Grenoble. After that, he spend one year as a consultant engineer in a computer vision industry-Hyderabad, India. His research interests are geometric computer vision, robotics, abstract algebra, Geometric AI and mathematical optimization. He received best algorithm award from Disney Research for his work on "Multi-body Non-Rigid Structure from Motion" in NRSFM Challenge at CVPR 2017, Hawaii, USA.

Luc Van Gool is a Full Professor at ETH Zürich and KU Leuven. He studied Electrical Engineering at the University of Leuven in Belgium. In 1991 he received his Ph.D. degree from the same university, with a dissertation on the use of invariance in computer vision. In 1991 he became an assistant professor in Leuven and in 1996 professor. He still leads a research group in Leuven, that focuses on industrial applications of computer vision. In 1998 he became a full professor at the ETH Zürich, where he is the head of Computer Vision Lab in the Department of Information Technology and Electrical Engineering. With his research teams, Luc Van Gool is a partner in several national and international projects, e.g. the EU ACTS project Vanguard, the EU Esprit projects Improofs and Impact, and the EU Brite-Euram project Soquetec. He is also involved in several other projects, that range from fundamental research to application-driven developments. His major research interests include 3D Vision, 2D and 3D object recognition, texture analysis, range acquisition, stereo vision, robot vision, and optical flow. Luc Van Gool has been a member of the program committees of several leading international conferences, including the ICCV, ECCV, and CVPR. In 1998 he received a David Marr Prize at the International Conference on Computer Vision. He is also a cofounder and director of the company Eyetronics, that specialises on 3D modeling and animation, mainly for the entertainment industry and medical applications. The "ShapeSnatcher" product received one of the EU EITP prizes in 1998. He has received several awards for his work including FWO Excellence Prize 2015 and, IEEE Computer Society Distinguished Researcher award at ICCV 2017.

Carlos Eduardo Porto de Oliveira Graduated with Bachelor in Social Communication with Film specialization. He is producing and directing documentaries, advertising films and all types of audiovisual content for different brands around the world. Much of his career was spent in Brazil, where he had the opportunity to specialize in animation and special effects. He was working for television channels in Brazil like MTV and ESPN, and also for biggest musical festivals developing all the content for big events. In Switzerland, he went through film studios, animation and film production companies, developing not only films but audio-visual and interactive installations. Currently, he works at ETH-CVL, where he is responsible for the audio-visual communication and special effects implementation, specifically directed to the most important scientific research in the CVL lab.

Anoop Cherian is a Research Scientist with Mitsubishi Electric Research Labs (MERL) Cambridge, MA and an Adjunct Researcher affiliated to the Australian Centre for Robotic Vision (ACRV) at the Australian National University. Previously, he was a Postdoctoral Researcher in the LEAR team at INRIA at Grenoble. He received his B.Tech (honors) degree in computer science and Engineering from the National Institute of Technology, Calicut, India in 2002, his M.S. and Ph.D. degrees in computer science from the University of Minnesota, Minneapolis in 2010 and 2013 respectively. His research interests lie in the areas of computer vision and machine learning.

Yuchao Dai is a Professor in School of Electronics and Information at Northwestern Polytechnical University, Xi'an, China. He was an ARC DECRA Fellow with the Research School of Engineering at the Australian National University, Canberra, Australia. He received the B.E. degree, M.E degree and Ph.D. degree all in signal and information processing from Northwestern Polytechnical University in 2005, 2008 and 2012, respectively. His research interests include structure from motion, multiview geometry, deep learning, compressive sensing and optimization. He won the Best Paper Award in CVPR 2012, DSTO Best Fundamental Contribution to Image Processing Paper Prize in 2014 and Best Algorithm award in CVPR NRSFM Challenge 2017. He served as an Area Chair for WACV 2019/2020 and ACM MM 2019.

Hongdong Li is a Chief Investigator of the Australia Centre of Excellence for Robotic Vision, Australian National University. He is Associate Director (for Research) for ANU School of Engineering. He was a visiting professor with the Robotics Institute, CMU during sabbatical in 2017. His research interests include geometric computer vision, pattern recognition and machine learning, vision perception for autonomous driving, and combinatorial optimization. He is an Associate Editor for IEEE TPAMI, and served as Area Chair for recent years' CVPR, ICCV and ECCV. He was the winner of CVPR Best Paper Award 2012, Marr Prize (Honorable Mention) at ICCV 2017, IEEE ICPR and IEEE ICIP Best Student Paper Winner, DSTO Best Fundamental Contribution to Image Processing Paper Prize at DICTA 2014 and Best algorithm award in CVPR NRSFM Challenge 2017. He is a program co-chair for ACCV 2018 and ACCV 2022. His research is funded by Australia Research Council, CSIRO, General Motors, Ford Motors, Microsoft Research etc. He is Presentation Co-Chair for ICCV 2019 and AC for CVPR 2020 and ECCV 2020.